# ZigBee OEM Module

# ProBee-ZE10

# User Guide

**Sena Technologies, Inc.**

**Rev 1.1**

**ProBee-ZE10 User Guide**
Firmware version PTv1.X

**Copyright**
Copyright 2010 Sena Technologies, Inc. All rights reserved.
Sena Technologies reserves the right to make changes and improvements to its product without providing notice.

**Trademark**
ProBee™ is a trademark of Sena Technologies, Inc.
Windows® is a registered trademark of Microsoft Corporation.
Ethernet® is a registered trademark of XEROX Corporation.

**Notice to Users**
When a system failure may cause serious consequences, protecting life and property against such consequences with a backup system or safety device is essential. The user agrees that protection against consequences resulting from system failure is the user's responsibility.
This device is not approved for life-support or medical systems.
Changes or modifications to this device not explicitly approved by Sena Technologies will void the user's authority to operate this device.

**Technical Support**
Email: support@sena.com
Website: http://www.sena.com

# Contents

**1 INTRODUCTION ..... 9**

1.1 About This Document ..... 9
1.2 Overview ..... 9

**2 GETTING STARTED ..... 11**

2.1 Hardware Installation ..... 11
*2.1.1 Development Board Layout ..... 11*
*2.1.2 Antenna Installation ..... 12*
*2.1.3 Module Installation on the Development Board ..... 12*
*2.1.4 Serial/USB Cable Installation ..... 13*
*2.1.5 Powering Development Board ..... 14*
*2.1.6 USB Driver Installation ..... 14*
2.2 Using Terminal Software for AT Commands ..... 15
2.3 Using ProBee Manager for Configuration ..... 17

**3 ZIGBEE NETWORK CONFIGURATION ..... 18**

3.1 Setting up ZE10 as a Coordinator ..... 18
*3.1.1 Node Type Selection for Coordinator ..... 19*
*3.1.2 Channel Mask Configuration for Coordinator ..... 19*
*3.1.3 PAN ID Configuration for Coordinator ..... 20*
*3.1.4 Permit Joining ..... 20*
3.2 Setting up ZE10 as a Router ..... 21
*3.2.1 Node Type Selection for Router ..... 21*
*3.2.2 Channel Mask Configuration for Router ..... 22*
*3.2.3 PAN ID Configuration for Router ..... 22*
*3.2.4 Permit Joining ..... 22*
3.3 Setting up ZE10 as an End Device ..... 23
3.4 Setting up ZE10 as a Sleepy End Device ..... 23
3.5 ZigBee Security ..... 25
*3.5.1 Security Level ..... 26*
*3.5.2 Link Key ..... 26*
*3.5.3 Network Key ..... 26*
*3.5.4 Update Key ..... 26*

**4 DATA TRANSMISSION ..... 27**

4.1 Command Mode ..... 27
*4.1.1 Unicast ..... 27*

*4.1.2 Multicast ..... 27*
*4.1.3 Broadcast ..... 28*
*4.1.4 Sending Binary Data ..... 28*
*4.1.5 Display Incoming Message ..... 28*
4.2 Data Mode ..... 29
*4.2.1 Unicast Data Mode ..... 29*
*4.2.2 Multicast Data Mode ..... 31*
*4.2.3 Broadcast Data Mode ..... 33*
**5 DIGITAL AND ANALOG I/O ..... 35**
5.1 GPIO Configuration ..... 35
5.2 I/O Sampling ..... 36
*5.2.1 Queried Sampling ..... 36*
*5.2.2 Periodic Sampling ..... 38*
*5.2.3 Change Detection Sampling ..... 38*
*5.2.4 GPIO Remote Control ..... 39*
**6 WORKING WITH DEVELOPMENT KIT ..... 40**
6.1 Development Kit Contents ..... 40
6.2 Development Board Features ..... 40
6.3 Development Board Hardware Descriptions ..... 41
*6.3.1 Power ..... 41*
*6.3.2 RS232 Port ..... 42*
*6.3.3 USB Port ..... 42*
*6.3.4 ZE10 GPIO Interface ..... 42*
*6.3.5 LEDs (CR12-19) ..... 43*
*6.3.6 Button Switches (SW10-17) ..... 43*
*6.3.7 Variable Resistors (VR1, VR2) ..... 43*
*6.3.8 Temperature Sensor ..... 43*
*6.3.9 Light Sensor ..... 43*
*6.3.10 Terminal Block ..... 44*
6.4 Configuration Example #1: 1 Coordinator, 1 Router and 1 End-Device ..... 44
*6.4.1 Coordinator Configuration ..... 45*
*6.4.2 Router Configuration ..... 45*
*6.4.3 End-device Configuration ..... 46*
*6.4.4 Monitoring the Sensors ..... 46*
6.5 Configuration Example #2: 1 Coordinator and 2 End-Devices ..... 46
*6.5.1 Coordinator Configuration ..... 47*
*6.5.2 End-devices Configuration ..... 47*
*6.5.3 Turning on/off the LEDs ..... 48*

**7 AT COMMAND REFERENCE ..... 49**

7.1 Command Line Format ..... 49
7.2 Standard AT Commands ..... 49
*7.2.1 AT ..... 49*
*7.2.2 ATB ..... 49*
*7.2.3 ATD ..... 49*
*7.2.4 ATE ..... 50*
*7.2.5 ATM ..... 50*
*7.2.6 ATO ..... 50*
*7.2.7 ATR ..... 50*
*7.2.8 ATS ..... 50*
*7.2.9 ATZ ..... 50*
*7.2.10 AT&F ..... 50*
*7.2.11 +++ ..... 51*
7.3 Node Specific Information ..... 51
*7.3.1 AT+LONGADDR or AT+LA ..... 51*
*7.3.2 AT+SHORTADDR or AT+SA ..... 51*
*7.3.3 AT+ADDRDISCOVERY or AT+AD ..... 51*
*7.3.4 AT+OPCH or AT+OC ..... 51*
*7.3.5 AT+OPPANID or AT+OI ..... 52*
*7.3.6 AT+OPEPID or AT+OE ..... 52*
*7.3.7 AT+VERSION or AT+VR ..... 52*
*7.3.8 AT+PRODUCTNAME or AT+PN ..... 52*
7.4 Network Specific Parameters ..... 52
*7.4.1 AT+CHMASK or AT+CM ..... 52*
*7.4.2 AT+PANID or AT+PI ..... 53*
*7.4.3 AT+EPID or AT+EI ..... 53*
*7.4.4 AT+GROUPID or AT+GI ..... 53*
*7.4.5 AT+POWER or AT+PW ..... 53*
*7.4.6 AT+STACK or AT+ZS ..... 53*
*7.4.7 AT+NODENAME or AT+NN ..... 54*
7.5 Network Formation and Join ..... 54
*7.5.1 AT+ASCAN or AT+AS ..... 54*
*7.5.2 AT+ESCAN or AT+ES ..... 54*
*7.5.3 AT+DSCAN or AT+DS ..... 54*
*7.5.4 AT+NODETYPE or AT+NT ..... 55*
*7.5.5 AT+PERMIT or AT+PJ ..... 55*
*7.5.6 AT+LEAVE or AT+LV ..... 55*
7.6 End-device ..... 55

*7.6.1 AT+SLEEP or AT+SM ..... 55*
*7.6.2 AT+PARENTLA or AT+PL ..... 55*
*7.6.3 AT+PARENTSA or AT+PS ..... 56*
*7.6.4 AT+CHILDTABLE or AT+CT ..... 56*
7.7 Data Transmission and Reception ..... 56
*7.7.1 AT+DESTLA or AT+DL ..... 56*
*7.7.2 AT+DESTGROUPID or AT+DG ..... 56*
*7.7.3 AT+SINKINFO or AT+SI ..... 56*
*7.7.4 AT+MAXPAYLOAD or AT+MP ..... 57*
*7.7.5 AT+TRASNSMITMODE or AT+TM ..... 57*
*7.7.6 AT+UNICAST or AT+UC ..... 57*
*7.7.7 AT+MULTICAST or AT+MC ..... 57*
*7.7.8 AT+BROADCAST or AT+BC ..... 57*
*7.7.9 AT+REMOTE or AT+RC ..... 57*
7.8 UART ..... 58
*7.8.1 AT+BAUDRATE or AT+UB ..... 58*
*7.8.2 AT+DATABIT or AT+UD ..... 58*
*7.8.3 AT+PARITY or AT+UP ..... 58*
*7.8.4 AT+STOPBIT or AT+US ..... 58*
*7.8.5 AT+FLOWCTRL or AT+UF ..... 59*
7.9 Security ..... 59
*7.9.1 AT+SECURITY or AT+SE ..... 59*
*7.9.2 AT+LINKKEY or AT+LK ..... 59*
*7.9.3 AT+NWKKEY or AT+NK ..... 59*
*7.9.4 AT+UPDATEKEY or AT+UK ..... 59*
7.10 GPIO ..... 60
*7.10.1 AT+GPIO or AT+IO ..... 60*
*7.10.2 AT+DIO ..... 60*
*7.10.3 AT+AI ..... 60*
*7.10.4 AT+ SOURCELA or AT+SL ..... 60*
7.11 Firmware Upload and Help ..... 61
*7.11.1 AT+BOOTLOAD or AT+BL ..... 61*
*7.11.2 AT+HELP or AT+HP ..... 61*
**8 AT COMMAND EXAMPLES ..... 62**
8.1 Network configuration ..... 62
*8.1.1 Coordinator ..... 62*
*8.1.2 Router ..... 62*
*8.1.3 End-device ..... 62*
*8.1.4 Sleepy End-device ..... 62*

*8.1.5 Specifying Network ..... 62*
8.2 Data Transmission ..... 63
*8.2.1 Command Mode ..... 63*
*8.2.2 Data Mode ..... 63*
8.3 UART Settings ..... 64
8.4 Security ..... 65
8.5 GPIO ..... 65
8.6 Firmware Upload ..... 65
*8.6.1 Local Node ..... 65*
*8.6.2 Remote Node ..... 66*
*8.6.3 Clone ..... 66*
**9 S-REGISTERS ..... 67**
9.1 Display Options ..... 67
*9.1.1 S11 ..... 67*
*9.1.2 S12 ..... 68*
*9.1.3 S13 ..... 68*
*9.1.4 S14 ..... 68*
9.2 Network Settings ..... 68
*9.2.1 S21 ..... 68*
*9.2.2 S22 ..... 69*
*9.2.3 S23 ..... 69*
9.3 Data Transmissions ..... 69
*9.3.1 S31 ..... 69*
*9.3.2 S32 ..... 69*
*9.3.3 S33 ..... 70*
*9.3.4 S34 ..... 70*
*9.3.5 S35 ..... 70*
*9.3.6 S36 ..... 71*
*9.3.7 S37 ..... 71*
*9.3.8 S38 ..... 71*
*9.3.9 S39 ..... 71*
9.4 GPIO ..... 72
*9.4.1 S41 ..... 72*
*9.4.2 S42 ..... 72*
*9.4.3 S43 ..... 72*
*9.4.4 S44 ..... 72*
*9.4.5 S45 ..... 73*
9.5 End Device ..... 73
*9.5.1 S51 ..... 73*

*9.5.2 S52 ..... 73*
*9.5.3 S53 ..... 73*
*9.5.4 S54 ..... 74*
*9.5.5 S55 ..... 74*
*9.5.6 S56 ..... 74*
9.6 Sink ..... 75
*9.6.1 S61 ..... 75*
*9.6.2 S62 ..... 75*
**10 PIN ASSIGNMENTS ..... 76**
**11 MECHANICAL DRAWINGS ..... 78**
**12 WARRANTY ..... 79**
12.1 GENERAL WARRANTY POLICY ..... 79
12.2 LIMITATION OF LIABILITY ..... 79
12.3 HARDWARE PRODUCT WARRANTY DETAILS ..... 79
12.4 SOFTWARE PRODUCT WARRANTY DETAILS ..... 80
12.5 THIRD-PARTY SOFTWARE PRODUCT WARRANTY DETAILS ..... 80
**13 REGULATORY INFORMATION ..... 81**
13.1 FCC ..... 81
13.2 CE ..... 81
13.3 TELEC ..... 82
13.4 KCC ..... 82
13.5 ZigBee Alliance ..... 82
**14 RF INFORMATION ..... 83**

# 1 Introduction

## 1.1 About This Document

This document provides an introduction on configuration and operation of the ProBee-ZE10 ZigBee OEM module. This document assumes the user is using the ZE10 starter kit for evaluation hence the pictures and configuration examples shown in this document are all based on the ZE10 starter kit.

This document does not provide full detail of the hardware specifications. Please refer to the ProBee-ZE10 data sheet for further references on the hardware design.

## 1.2 Overview

The ProBee-ZE10 is an all-in-one ZigBee OEM module with integrated ZigBee core and radio/antenna circuits together with high level software library. Using the ZE10, OEM manufacturers can easily and cost-effectively integrate ZigBee functionality into target products in timely manner.

The ZE10 OEM module is designed to meet specific requirements of low cost and low power wireless applications such as home automation or smart energy by adopting the ZigBee technology. The ZE10 is based on the Ember EM250 ZigBee core and fully certified under the latest ZigBee Pro standard for the maximum compatibility.

Despite of its low cost and low power consumptions, the ZE10 provides high performance and robust data transfer capability for wide range of applications. The ZE10 can transfer data at up to 250 kbps and reach up to 1.6 km (0.99 miles) outdoor line-of-sight.

**Key Features**

- Integrated 2.4GHz, IEEE 802.15.4-compliant transceiver
- ZigBee Pro Certified
- ZigBee core: Ember EM250
- Transmit Power: +20dBm E.I.R.P (max.)
- Receiver Sensitivity: -102dBm @1% BER
- TX Current: 190mA @3.3V (max.)
- RX Current: 45mA @3.3V (max.)
- Sleep Current: ≤ 2uA
- Working Distance:1.6 km (0.99 miles)
- Various Antenna/Connector Options: Dipole 1/3/5dBi, U.FL, Chip
- Full UART Signals Support: UART_TXD/RXD, RTS/CTS, DTR/DSR
- 4 Analog Inputs
- 13 Digital Inputs/Outputs
- RoHS Compliant

**Applications**

- Advanced Metering Infrastructure
- Home Area Networks (HAN)
- Neighborhood Area Networks (NAN)
- Home Automation
- Advanced lighting, entertainment, and climate control systems
- Service-based monitoring, security, and awareness systems
- Commercial Building Automation
- Climate & lighting control systems
- Industrial and Domestic Applications

# 2 Getting Started

Creating a ZigBee network using the ZE10 modules consists of the following steps:

- Install the ZE10 module on the development board
- Create a ZigBee network by configuring a ZE10 module as the coordinator
- Configure ZE10 modules as routers. This step is optional since the coordinator works as a router too.
- Configure ZE10 modules as end devices
- Transmit data between ZE10 devices

This chapter describes how to install the ZE10 modules on to the development boards and how to use terminal software programs to access the ZE10 modules using AT commands for configurations and operations. The rest of the steps including ZigBee network creation, coordinator/router/end-device configurations and data transmissions are described in the following chapters. The more detailed information on the development kit and the configuration examples are also provided in the later chapter.

## 2.1 Hardware Installation

### 2.1.1 Development Board Layout

Figure 2-1 shows the layout of the development board. The development board provides the interface between the ZE10 and the external serial port or USB port together with input/output terminals for convenient tests.

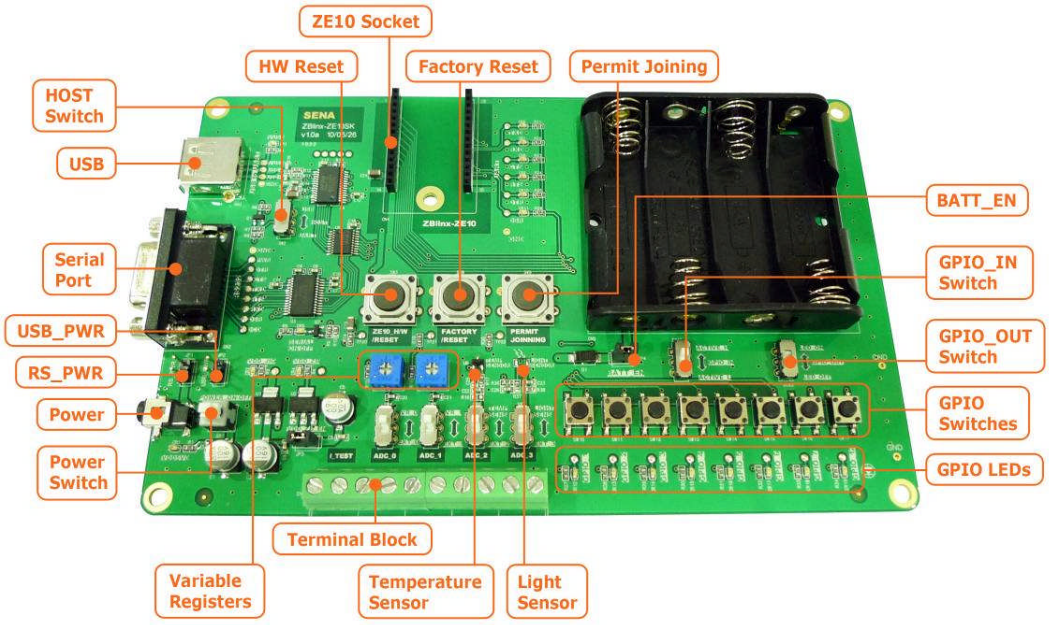



*Figure 2-1 Development Board Layout*

### 2.1.2 Antenna Installation

The ZE10 starter kit includes the several antennas for evaluation purposes. You can try different antennas for testing purpose and select the appropriate antenna for the actual applications. Figure 2-2 shows how to install the RP-SMA dipole antenna to the ZE10 module.

*Figure 2-2 RP-SMA Dipole Antenna Installation*

### 2.1.3 Module Installation on the Development Board

Once the antenna is installed on the ZE10 module, the ZE10 module can be installed on the development board.

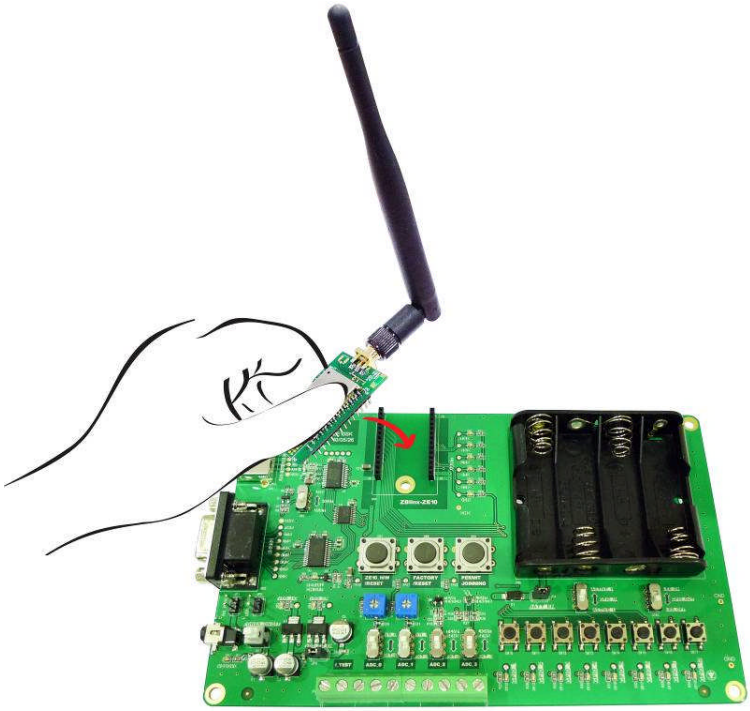


*Figure 2-3 ZE10 Installation on the Development Board*

### 2.1.4 Serial/USB Cable Installation

After the ZE10 is securely installed on the development board, please connect the included serial cable between the development board and your computer so AT commands can be sent from the computer to the ZE10 module. The serial port of the development board is DB9 male as a DTE terminal, and the serial cable included is a cross-over cable for DTE-DTE connection. To use the serial port, the HOST (SW2) switch should be set to UART.

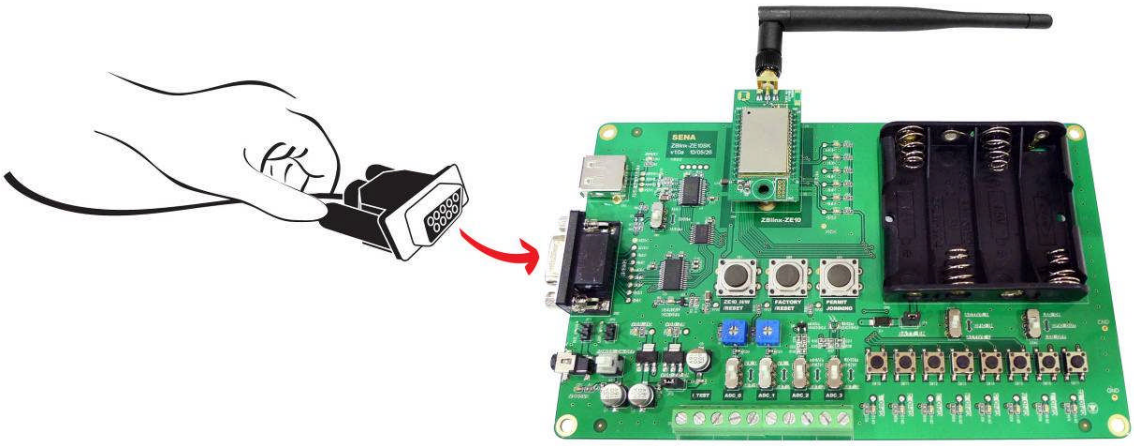


*Figure 2-4 Serial Cable Installation*

In case the serial port is not available for the computer, you can use the USB port of the development board instead. In this case, the internal USB-RS232 converter of the development board will convert the USB signal to the RS232 signal and vice versa. To use the USB port, the HOST (SW2) switch should be set to USB and the software driver for the USB in the CD should be installed on the computer. For the USB driver installation, please refer to the section 2.1.6.

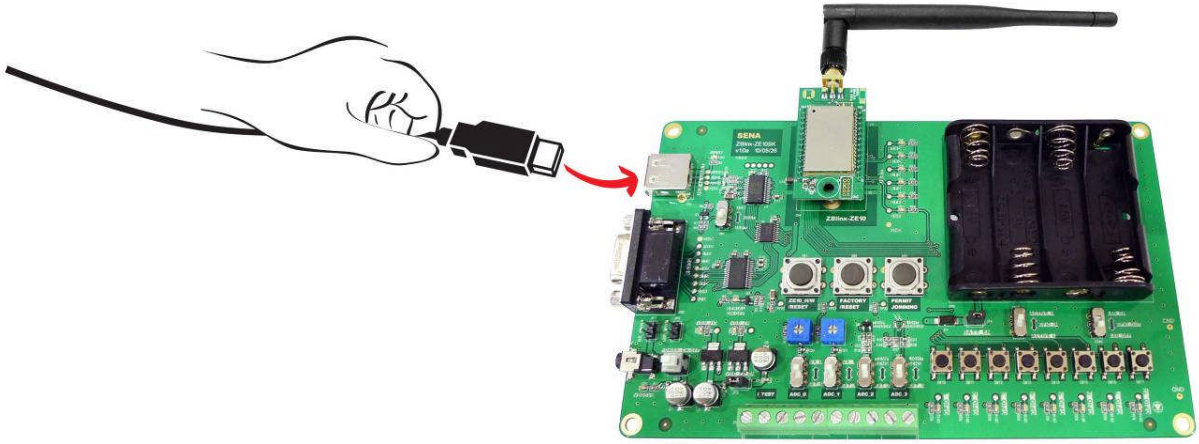


*Figure 2-5 USB Cable Installation*

### 2.1.5 Powering Development Board

Once all the installations are completed, please plug in the power connector using the included DC power adapter. The RS_PWR, USB_PWR and BATT_EN jumpers should be all OPEN to use the DC power adapter. Alternatively, the development board can be powered using batteries, or from the USB port or through the RS232 port pin 9. In these cases, the power source jumpers should be set accordingly. For details about the power source jumper settings, please refer to the section ***6.3.1 Power***.

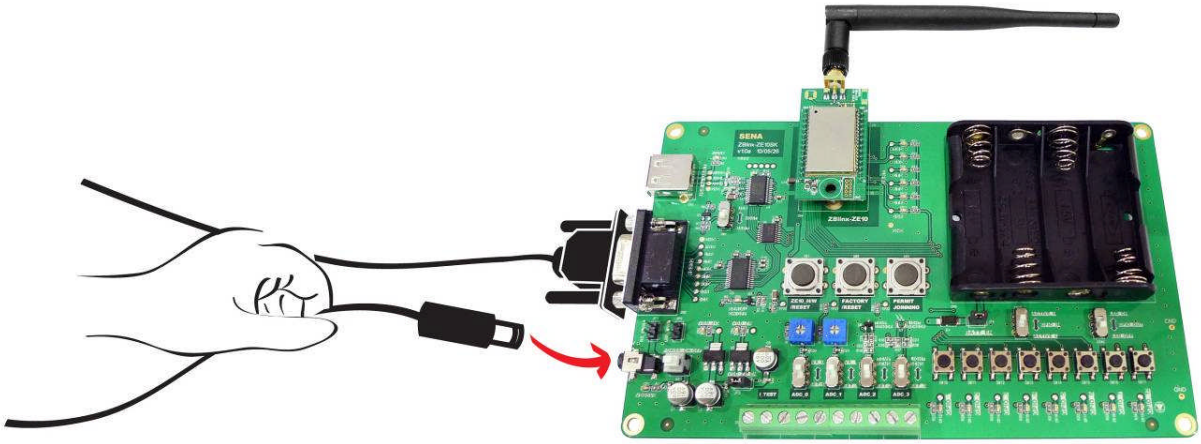


*Figure 2-6 Plugging DC Power Adapter*

### 2.1.6 USB Driver Installation

In case the USB interface is used for the communication between the development board and the computer, the corresponding software driver needs to be installed on the computer. Windows XP and later versions already include the USB driver. When Windows asks for the location of the USB driver, you can select the option to “install the software automatically” as shown in Figure 2-7. You can also choose to install the driver from the CD included if the driver is not installed correctly for some reason.

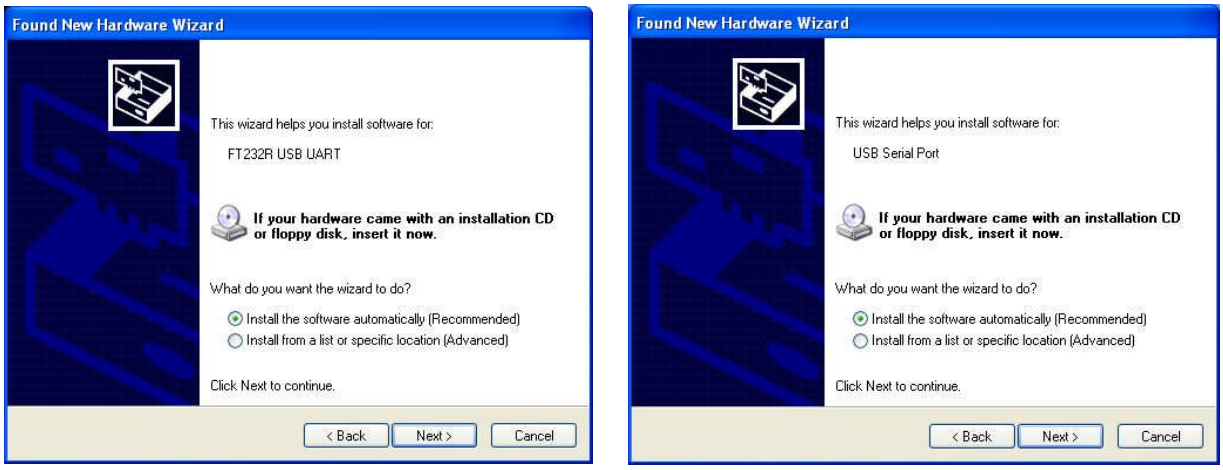

Found New Hardware Wizard
This wizard helps you install software for:
FT232R USB UART
If your hardware came with an installation CD or floppy disk, insert it now.
What do you want the wizard to do?
Install the software automatically (Recommended)
Install from a list or specific location (Advanced)
Click Next to continue.
< Back
Next >
Cancel
Found New Hardware Wizard
This wizard helps you install software for:
USB Serial Port
If your hardware came with an installation CD or floppy disk, insert it now.
What do you want the wizard to do?
Install the software automatically (Recommended)
Install from a list or specific location (Advanced)
Click Next to continue.
< Back
Next >
Cancel

*Figure 2-7 USB Driver Installation*

## 2.2 Using Terminal Software for AT Commands

Once the hardware installation is complete, the ZE10 unit is ready to communicate with the computer using the AT commands using a terminal emulation software. You can use any kind of terminal emulation software. This document assumes HyperTerminal is used. If you are using Windows Vista or 7, HyperTerminal is no longer included in the Windows so you need to download HyperTerminal from the Internet. For more details on HyperTerminal, please refer to Microsoft web page.

Once the HyperTerminal is installed, launch HyperTerminal and select the serial port that is connected to the ZE10 development board and enter the serial settings to use. The default serial settings of the ZE10 are **9600 bps, 8 data bit, none parity, 1 stop bit, hardware flow control**.

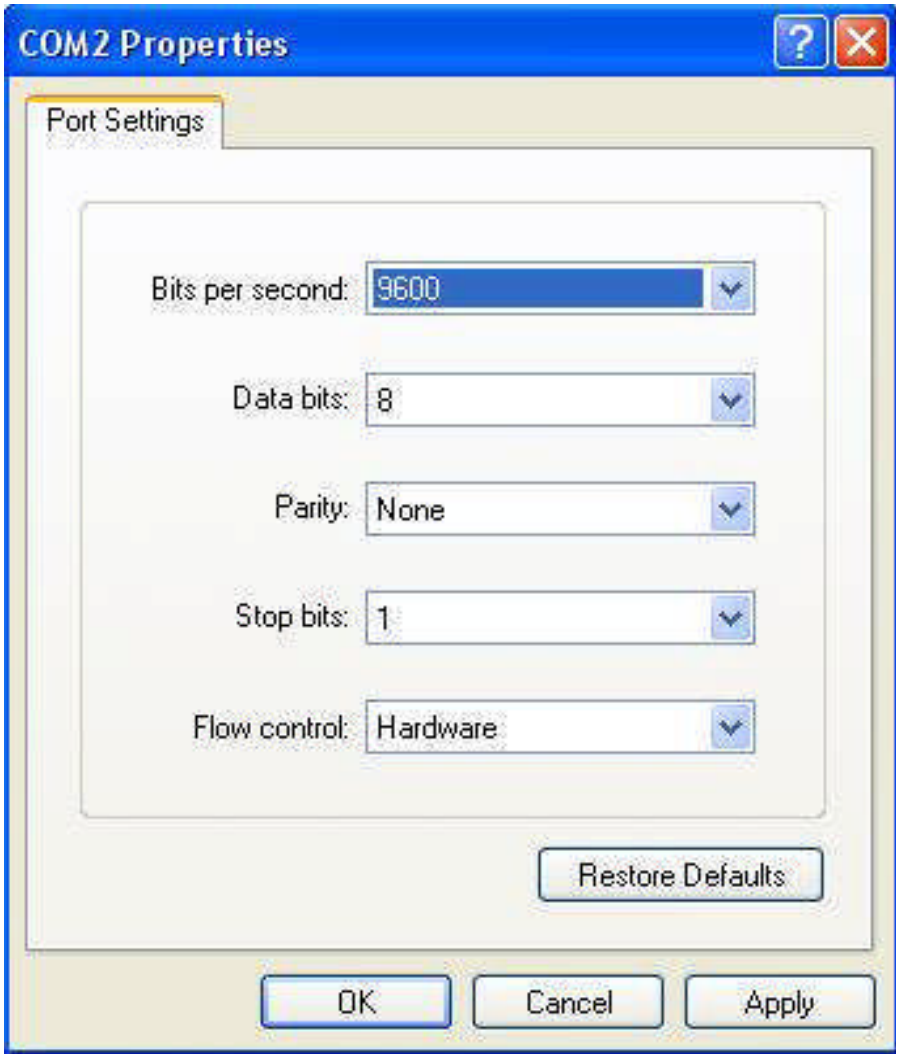



*Figure 2-8 HyperTerminal Serial Settings*

To display the AT commands that are being typed, you need to enable the local echo option on the HyperTerminal. To enable this option, Go to File->Properties->Settings->ASCII setup and select the "Echo typed characters locally" option. To verify the connection, type AT and press the Enter key. If the AT command is accepted by the ZE10, OK string will be displayed on the screen.

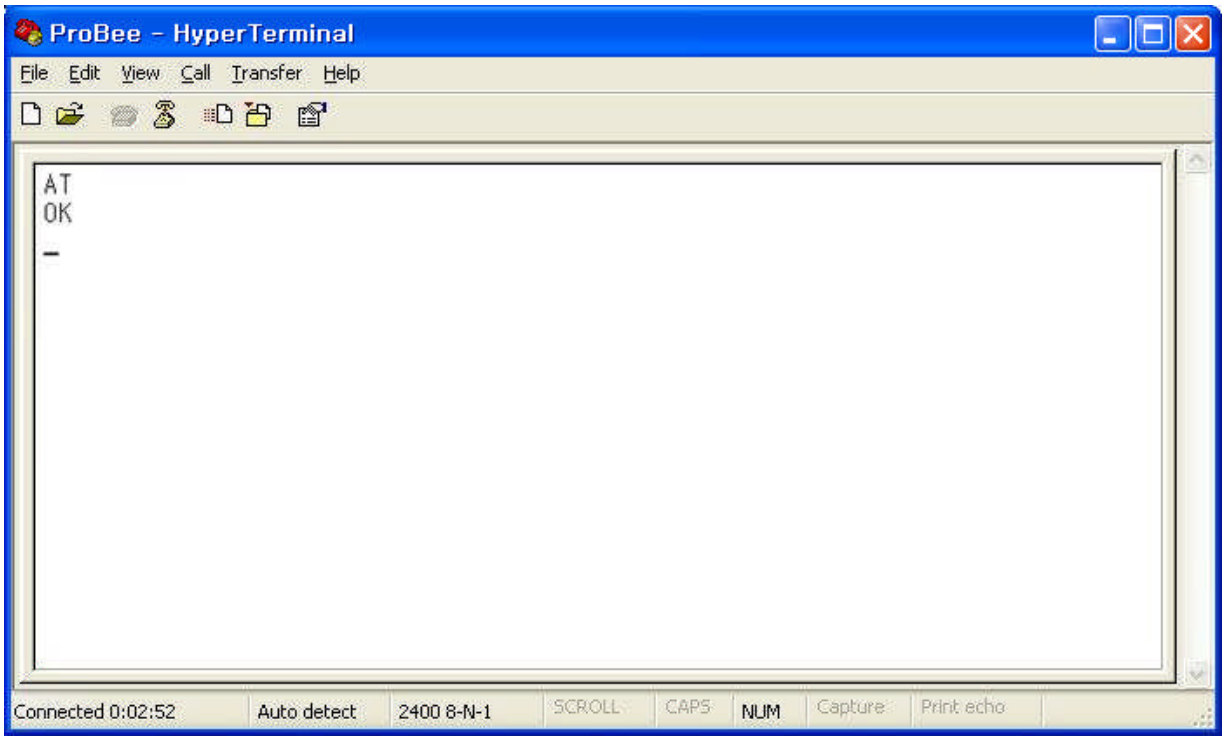



*Figure 2-9 Testing AT Command on HyperTerminal*

## 2.3 Using ProBee Manager for Configuration

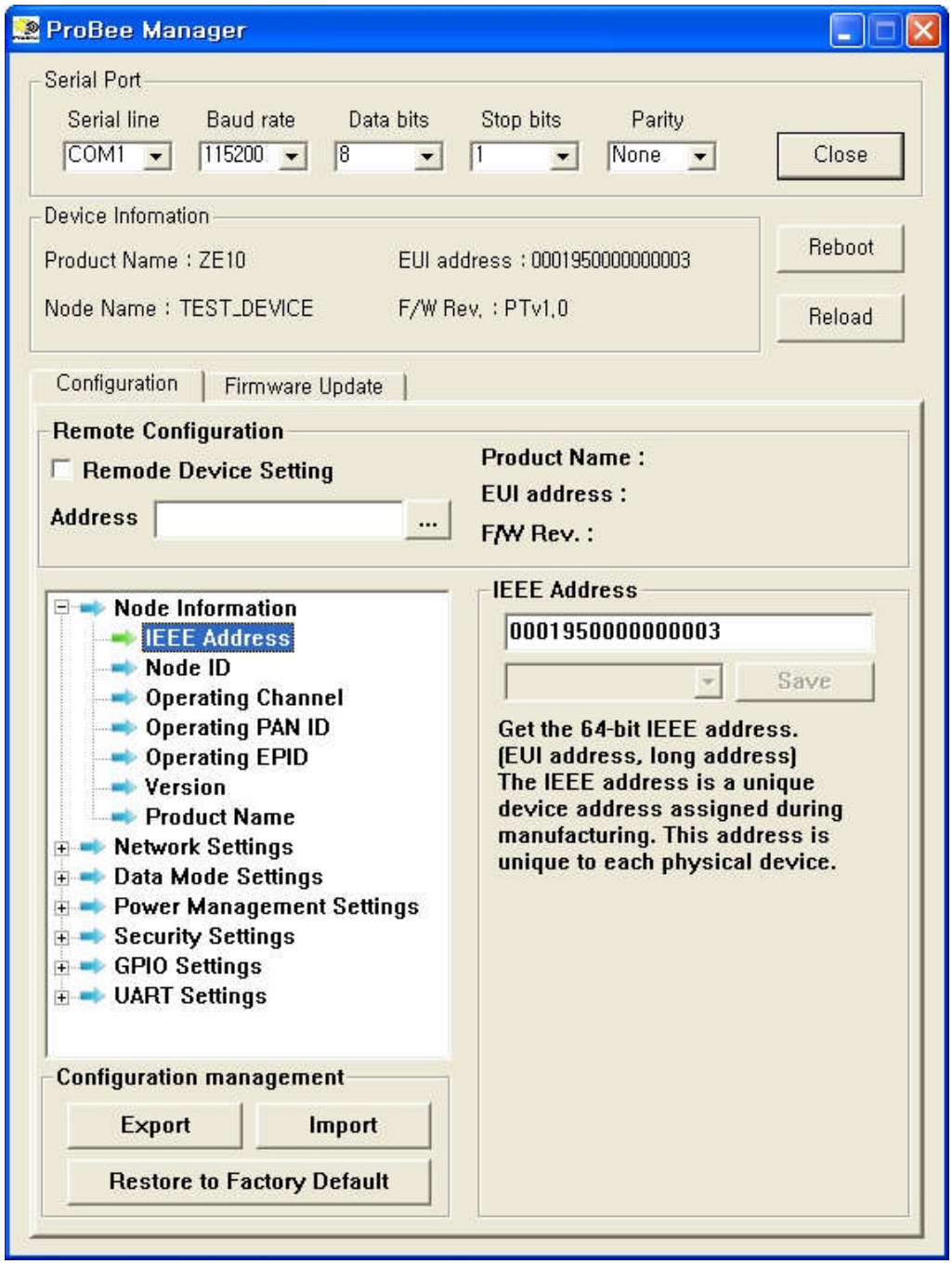



*Figure 2-10 ProBee Manager*

It is possible to set or get the configurations of the ZE10 using ProBee Manager. This PC utility helps that a user can configure several setting values easily without a terminal program and upload the firmware file to local and/or remote node.

# 3 ZigBee Network Configuration

A ZigBee Network consists of a coordinator, routers and end devices. A minimal ZigBee network consists of one coordinator and multiple end devices which directly connect to the coordinator. For larger ZigBee networks, routers are required to provide redundant routings to form mesh networks.

- **ZigBee Coordinator**: The coordinator forms the root of the network tree and might bridge to other networks. There is exactly one ZigBee coordinator in each network since the coordinator starts the ZigBee network itself. It also stores information about the network, including acting as the Trust Center & repository for security keys.
- **ZigBee Router**: The routers are able to route messages between other devices. Since the coordinator provides the routing functionality too, the routers are optional to form a ZigBee network.
- **ZigBee End Device**: The end device contains just enough functionality to talk to the parent node which is either the coordinator or a router. It cannot relay data from other devices.
- **ZigBee Sleepy End Device**: The sleepy end device is the same with normal end device except it can stay in the sleep mode much longer. This relationship allows the end device to sleep significant amount of the time and achieve the long battery life.

The ZE10 can be configured to work either as a coordinator, a router or an end device. Once the operation mode of the ZE10 is selected, corresponding parameters need to be set up appropriately. This chapter explains how to configure the ZE10 operation mode and other parameters related to the ZigBee network itself.

*Table 3-1 ZigBee Node Types*

| Node Type | Recommendation when to use ZE10 |
|---|---|
| Coordinator | It should be selected when it is first time to establish the network. |
| Router | It is recommended when there is a network already and it needs to be expanded. A router is suitable for fast and stable data transmissions and receptions. |
| End Device | It is recommended when it doesn't need to expand the network anymore and there are enough nodes as parents. An end-device consumes less power than a router. |
| Sleepy End Device | It is recommended when the data transmissions occur rarely and in very small quantities. A sleepy end device can lose data if sleep duration is too long, but it is possible to consume extremely low power as low as 2uA. |

## 3.1 Setting up ZE10 as a Coordinator

A coordinator relays messages and can act as a parent to other nodes. Every personal area network (PAN) must be started by a node acting as the coordinator. This device is normally always powered on.
To set up the ZE10 to work as the coordinator, please follow the steps as below

- Configure the ZE10 node type as the coordinator
- Optionally, select the channel mask. Otherwise, the ZE10 will use the default channel mask.
- Optionally, set up the PAN ID and/or the extended PAN ID. Otherwise, the ZE10 will generate the PAN ID and/or the extended PAN ID automatically.
- Permit joining when other router or end device tries to join the ZigBee network.

### 3.1.1 Node Type Selection for Coordinator

The ZE10 module can be configured as a ZigBee coordinator by the AT command as below:

*AT+NODETYPE=n or AT+NT=n*
*Where n is 0 = none, 1 = coordinator, 2 = router, 3 = end-device, 4 = sleepy end-device*

After the operation mode is selected, ATZ command should be performed to reset the unit to have the change applied.

Example)

```
AT+NODETYPE=1<CR>       # Set node type as the coordinator
OK<CR>                  # Response
ATZ<CR>                 # Apply the change
OK<CR>                  # Response
```

### 3.1.2 Channel Mask Configuration for Coordinator

When the ZigBee coordinator starts, it scans predefined frequency channels to find the best frequency channel to use without causing interference with other wireless devices. The user is able to specify the channels to use with the AT command as below:

*AT+CHMASK=n*
*Where n = channel mask to use. Default = 0x03FFF000*

This value has a 32-bit hexadecimal unsigned integer. Each bit represents whether a channel will be used or not. For example, to use a channel between 12 and 15, the 12, 13, 14 and 15 bits of the channel mask should be 1 and all other bits should be 0. The default setting is 0x3FFF000 which means that all the channels from 12 to 25 can be used. The ZE10 supports only the channels between 12 and 25, so 0~11 bits and 26~31 bits should be 0. Table 3-2 shows the bit mask for each channel.

*Table 3-2 Channel Bitmask*

| Channel | 12 | 13 | 14 | 15 | 16 | 17 | 18 |
|---|---|---|---|---|---|---|---|
| Bit mask | 00001000 | 00002000 | 00004000 | 00008000 | 00010000 | 00020000 | 00040000 |
| Channel | 19 | 20 | 21 | 22 | 23 | 24 | 25 |
| Bit mask | 00080000 | 00100000 | 00200000 | 00400000 | 00800000 | 01000000 | 02000000 |

### 3.1.3 PAN ID Configuration for Coordinator

Every ZigBee network should be assigned a PAN ID to identify the network. If the PAN ID is 0000, the ZE10 will generate a random 16-bit PAN ID to start the ZigBee network. Otherwise, the user can enter the PAN ID using the AT command as below:

*AT+PANID=xxxx*
*Where xxxx is PAN ID to use. For example, 12AB. Default setting is 0000*

When choosing a PAN ID, the user needs to make sure that there is no ZigBee network using the same PAN ID. If the PAN ID is already used by another ZigBee network, the PAN ID is unavailable and the user should choose a new PAN ID. In order to avoid this, PAN ID auto-generation is strongly recommended.

The user can also assign the 64-bit extended PAN ID to the ZE10 using the AT command below. If the extended PAN ID is 0000000000000000, the ZE10 uses its IEEE address as the extended PAN ID. The extended PAN ID cannot be shared with other ZigBee networks either.

*AT+EPID=xxx…xxx*
*Where xxx…xxx is EPID to use. For example, 0123456789ABCDEF. Default setting is 000…000.*

### 3.1.4 Permit Joining

When a router or an end device tries to join the ZigBee network, there should be at least one coordinator or router which is permitting joining. If there is no router in the ZigBee network yet, the coordinator should permit joining. The ZE10 provides three ways to permit joining are as follows:

- Permanently permitting by configuration
- Transiently permitting by AT command
- Transiently permitting by digital input signal

If joining is permanently permitted, any device can join anytime. Since this will allow even an unidentified device to join the network anytime, this mode should be selected with extra caution. The AT command to permit joining permanently is as follows:

*AT+PERMIT=255 or AT+PJ=255*

The user can permit joining for a moment at need. The AT command to permit joining is

*AT+PERMIT=n or AT+PJ=n*
*Where n is the duration to permit in seconds ranging 1~254 seconds.*

The user can also configure the ZE10 to permit joining when the digital input GPIO0 is active. The AT command to enable this functionality is as follows:

*AT+GPIO0=5 or AT+IO0=5*

When the ZE10 detects falling edge of the GPIO0, the ZE10 permits joining for the duration that is configured in S-register 22 (S22). The AT command to change the duration is as follows:

*ATS22=n*
*Where n is the duration to permit in seconds ranging 1~254 seconds. Default is 60.*

S22 value determines the time that the device will allow other devices to join to it. If set to 255, the device will always permit other nodes joining. If the user reboots the coordinator (or a router), the device allows other nodes to join the network for the period defined by S-register 22.

*Note: When a device that has joined a network reboots, the device is able to join the network regardless of permitting. But, after the device has joined other network, rejoining the previous network should be permitted by the coordinator or a router.*

## 3.2 Setting up ZE10 as a Router

A router is a full-function routing device that relays messages and can act as a parent to other nodes. These devices must be always powered on.

To configure the ZE10 as a ZigBee router, please follow the steps below.

- Configure the ZE10 node type as a router.
- Optionally, select the channel mask. Otherwise, the ZE10 will use the default channel mask.
- Optionally, set up the PAN ID and/or the extended PAN ID. Otherwise, the ZE10 will scan and join the ZigBee network found automatically.
- Permit joining when other router or end device tries to join the ZigBee network.

### 3.2.1 Node Type Selection for Router

The ZE10 module can be configured as a ZigBee router by the AT command as below:

*AT+NODETYPE=n or AT+NT=n*
*Where n is 0 = none, 1 = coordinator, 2 = router, 3 = end-device, 4 = sleepy end-device*

After the operation mode is selected, ATZ command should be performed to reset the unit to have the change applied.

Example)

```
AT+NODETYPE=2<CR>       # Select router mode
OK<CR>                  # Response
ATZ<CR>                 # Apply the change
OK<CR>                  # Response
```

### 3.2.2 Channel Mask Configuration for Router

When the ZE10 as a ZigBee router or a ZigBee end device tries to join, it scans all the channels configured by channel mask to find the ZigBee network to join. The channel mask of router or end device should include the channel on which the coordinator is operating. For example, we assume that the coordinator of the network to join is operating on channel 12. The router or end device trying to join should has a channel mask of which $12^{th}$ bit is 1

### 3.2.3 PAN ID Configuration for Router

The PAN ID of a ZigBee router or a ZigBee end device should be the same with the PAN ID of the ZigBee coordinator. If the coordinator is using a randomly generated PAN ID, the user can get the operating PAN ID by running the AT command below on the coordinator:

*AT+OPPANID*

If the PAN ID of a ZigBee router or a ZigBee end device is 0000, the PAN ID is ignored. After checking the PAN ID, the ZE10 trying to join compares the extended PAN ID to make sure whether the network is the right network to join. If the extended PAN ID is the same with its configured extended PAN ID, it will try to join the network. Otherwise, it will try to find another network. The user can get the operating extended PAN ID by running the AT command below on the coordinator:

*AT+OPEPID*

If the extended PAN ID of a ZigBee router or a ZigBee end device is 0000000000000000, the extended PAN ID is ignored

### 3.2.4 Permit Joining

The ZigBee router is configured and works the same way with the coordinator to permit joining of other nodes. Please refer to the section ***3.1.4 Permit Joining*** for the configuration.

## 3.3 Setting up ZE10 as an End Device

An end device is a leaf node which communicates only with its parent node and will not relay messages directly. To configure the ZE10 as a ZigBee end device, please follow the steps below.

- Configure the ZE10 node as an end device
- Optionally, select the channel mask. Otherwise, the ZE10 will use the default channel mask.
- Optionally, set up the PAN ID and/or the extended PAN ID. Otherwise, the ZE10 will scan and join the ZigBee network found automatically.
- Configure the poll timeout and the poll period

Please refer to the section ***3.2 Setting up ZE10 as a Router*** for the AT commands regarding the node type selection through the PAN ID configurations.

When an end device joins the network, the power LED blinks periodically if GPIO1 is set to specific functionality. The end device belongs to a parent node which is either the coordinator or a router, and the parent node registers the end device to its internal child table for message relaying. This table is maintained as long as the end device polls the parent node periodically within the poll timeout value of the parent node. If the end device does not poll for the poll timeout value of the parent node, the parent node removes the end device from its child table and does not relay the message any more. Hence, the poll period of the end device should be equal or shorter than the poll timeout of the parent node to avoid accidental removal of the end device from the parent node's child table.

The ZE10 poll interval can be configured by setting the S-registers using the AT commands as below.

*ATS51=n, where n is poll timeout in seconds and the default is 60.*

*ATS56=n, where n is poll period in seconds and the default is 5.*
End device polls the parent node at poll period defined by S56 periodically. It is for the connection to the parent node and data transmission with other nodes in the network.

Even though the end device needs to poll the parent node continuously, it does not go into the sleep mode and it can send and receive messages any time unlike the sleepy end device.

## 3.4 Setting up ZE10 as a Sleepy End Device

A sleepy end device can turn off radio to save power. To configure the ZE10 as a ZigBee sleepy end device, please follow the steps below.

- Configure the ZE10 node as a sleepy end device.
- Optionally, select the channel mask. Otherwise, the ZE10 will use the default channel mask.
- Optionally, set up the PAN ID and/or the extended PAN ID. Otherwise, the ZE10 will scan and join

the ZigBee network found automatically.

- Configure the settings for sleep interval.

Please refer to the section ***3.2 Setting up ZE10 as a Router*** for the AT commands regarding the node type selection through the PAN ID configurations.

One of the unique features of the ZigBee device is its extreme low power consumption which is mainly achieved by its efficient sleep and wake-up mechanism. While ZigBee coordinator and routers are required to be in operation always, ZigBee sleepy end devices can be configured to be in the sleep mode most of the time and only wake up at pre-configured time interval for data transmission before it goes into the sleep mode again. The ZE10 sleep interval can be configured by setting the S-registers using the AT commands as below.

*ATS51=n, where n is poll timeout in seconds and the default is 60.*
This value is for a parent node. If an end device does not send a poll request to its parent coordinator or router within the poll timeout, the end device is removed from the child table. Unlike the non-sleepy end device, sleepy end device does not use S51 as the poll period. Instead, it uses S53 as the sleep duration which works similar to the poll period, except it goes to sleep mode.

*ATS53=n, where n is sleep duration in quarter seconds and the default is 8.*
The ZE10 sleeps at sleep duration and turns off radio.

*ATS52=n, where n is poll period in milliseconds and the default is 100.*
Sleepy end device wakes up from the sleep mode if S53 timer expires or GPIO0 interrupt is generated or serial data is received. Once the sleepy end device wakes up, it polls the parent node at poll period defined by S52 until it goes to sleep mode again. Since the sleepy end device exchanges data with the parent node only by polling, S52 poll period is usually configured as around 100 ms.

*ATS54=n, where n is standby timeout in seconds and the default is 5.*
If there is no RF data or serial data for S54 seconds, the sleepy end device goes to sleep mode again. If any serial or RF data is sent or received, the timer will reset and the sleepy end device waits for another S54 seconds before it goes to sleep mode again.

*ATS55=n, where n is rejoin timeout and the default is 10.*
When the connection to the parent node is broken for S55 in seconds, the child node tries to rejoin the current network.

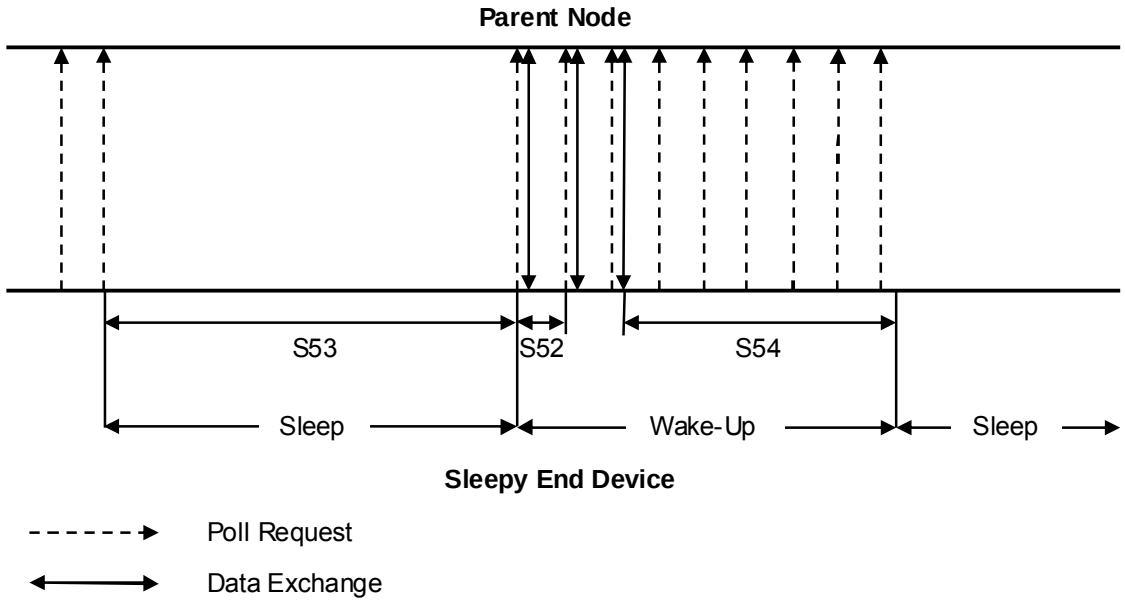



*Figure 3-1 Sleep Interval and Wake-up Timing*

The sleepy end device can also wake up from the sleep mode by GPIO0 interrupt or incoming serial data. Once it wakes up, it also polls with the parent node at S52 interval and goes into sleep mode after S54 time elapses after the last data exchange.

*Note: If the sleepy end device wakes up by incoming serial data, the integrity of the first incoming byte is not guaranteed. It is strongly recommended to wake up the sleepy end device by using GPIO0 first before sending the serial data or use higher level serial data protocol to check the data integrity.*

## 3.5 ZigBee Security

Security is a major concern in the ZigBee architecture. Although ZigBee uses the basic security elements in IEEE 802.15.4 (e.g., AES encryption & CCM security modes), the ZE10 increase the security level with:

- 128 bit AES encryption algorithms
- Strong, NIST-approved security
- Defined Key Types (Link, Network)
- Defined Key setup and maintenance
- CCM (Unified/Simpler mode of operation)
- Trust Centers

### 3.5.1 Security Level

In order to use security in the application, a user should set security level using the AT command below. Set 1 defines both Authentication and Encryption at the network layer. Set 0 defines no security. Disabling security in the application is not ZigBee compliant. All nodes in the network should have the same security level setup. The security level is configured using the AT command below.

*AT+SECURITY=n, where n is encryption level. (0=disable, 1=enable)*

### 3.5.2 Link Key

The trust center (normally coordinator in network) requests link key before a device is trying to join network. If link key the device has is wrong, joining process will be denied. Hence, all nodes in the network should have the same link key setup before joining the network.

*AT+LINKKEY=xxx…xxx, where xxx…xxx is 128-bit hexadecimal trust center link key. Default is FFF…FF.*

### 3.5.3 Network Key

It is the network-wide key used to secure transmissions at the Network Layer. The network key encrypts all transmissions at the Network Layer. The network key only needs to be set up for the coordinator.

*AT+NWKKEY=xxx…xxx, where xxx…xxx is 128-bit hexadecimal network key. Default is FFF…FF.*

### 3.5.4 Update Key

A trust center can update the Network Key to help minimize the risk associated with a particular instance of the network key being compromised. Key updates are broadcast by the trust center throughout the network, encrypted using the current network key. Devices that hear the broadcast will not immediately use the key, but simply store it. Later, a key switch is broadcast by the trust center to tell all nodes to start using the new key. At a minimum, the trust center should allow adequate time for the broadcast of the new key to propagate throughout the network before switching.

*AT+UPDATEKEY<CR>*

# 4 Data Transmission

The ZE10 supports three different types of data transmissions: unicast, multicast and broadcast. Unicast is to transmit data to a specific ZigBee device. Multicast is to transmit data to a group of ZigBee devices that are assigned a specific group ID. Broadcast is to transmit data to all ZigBee devices (or specific type of ZigBee nodes if specified) in the network.

There are also two different methods to transmit data to other device as below:

- Command mode: It sends only one packet up to allowed payload length by a form of command.
- Data mode: It is possible to send data consecutively until leave the data mode using "+++".

## 4.1 Command Mode

### 4.1.1 Unicast

Unicast transmissions are sent from a source device to a destination device. The destination device can be any ZigBee device in the ZigBee network. To specify the destination device, the ZigBee address of the destination device should be known.

There are two different ZigBee addresses assigned to each ZigBee device. The 64-bit address is a unique device address assigned during manufacturing. The 64-bit address is also called IEEE address or extended address. The 16-bit address is assigned to the device when it joins a ZigBee network. The ZE10 can send unicast messages based on either 16-bit address or 64-bit address. The AT command for the unicast transmissions is as below:

*AT+UNICAST=<node_id>,<MSG><CR>*
*Where <node_id> = 16-bit or 64-bit address, <MSG> = data to transmit*

The maximum payload of the message is 90 bytes (or 72 bytes under security enabled). If the message length exceeds the maximum payload, the message cannot transmit and the error message will be presented.

### 4.1.2 Multicast

Multicast transmissions are sent from a source device to destination devices with the specific group ID. The group ID is assigned to the devices using the AT+GROUPID command. The AT command for the multicast transmissions is as below.

*AT+MULTICAST=<group_id>,<MSG><CR>*
*Where <group_id> = group ID, <MSG> = data to transmit*

The data transmission range can be set using the AT commands below. Please refer to the ***9 S-Registers*** for more details.

*ATS36=n, where n is the radius.*
*ATS37=n, where n is the number of hops through nonmembers.*

### 4.1.3 Broadcast

Broadcast transmissions are sent from a source device to all devices in the ZigBee network. The AT command for the broadcast transmissions is as below.

*AT+BROADCAST=<MSG><CR>*
*Where <MSG> = data to transmit*

The user can also specify the transmit range by changing the S35 register value as below. Please refer to the ***9 S-Registers*** for more details.

*ATS35=n, where n is the broadcast range.*
*ATS36=n, where n is the radius.*

### 4.1.4 Sending Binary Data

There are some characters, e.g. carriage return, line feed and back space, that are not allowed or have special meanings in command mode. In order to transmit these characters in command mode, these characters must be encoded in a command line. The special characters encoding table is shown in Table 4-1.

*Table 4-1 Special Characters Encoding Table*

| HEX | Abbreviation | Glyph | Encoded | Description |
|---|---|---|---|---|
| 00 | NULL | - | \00 | Null character |
| 0D | CR | - | \0D | Carriage return |
| 0A | LF | - | \0A | Line feed |
| 08 | BS | - | \08 | Backspace |
| 09 | HT | - | \09 | Horizontal Tab |
| 7F | DEL | - | \7F | Delete |
| 5C | - | \ | \5C | Control character for binary encoding |

For example, when sending *abc<CR>* with AT+BROADCAST command, the command should be as follows:

*AT+BROADCAST=abc\0D<CR>*

### 4.1.5 Display Incoming Message

S11 register determines whether incoming message will be presented or not in the command mode. In

the data mode, on the other hand, incoming data is always displayed. To change the S11 register, please use the AT command as below:

*ATS11=<value><CR>,*
*where <value> = 0 to hide incoming message, 1 to show incoming message.*

If user wants to display incoming message in command mode, S11 should be enabled. It is presented by a form of +<*IEEE address*>|<MSG><CR>.

When displaying incoming messages in command mode, some characters are encoded in the same way with sending binary data as in Table 4-1. For example, when receiving *abc<CR>* from 0001950000000001, the displaying message is as follows:

*+0001950000000001| abc\0D<CR>*

## 4.2 Data Mode

If the data is too long to send using the command mode, Data Mode can be used instead. Once the ZE10 enters a Data Mode session, the ZE10 can send data continuously to a specific node or current network until the data mode is escaped. The data is packetized and transmitted at every inter-character timeout. This data link to the designated device is maintained until the escape command "+++" is entered. To return to the last data mode after the ZE10 is escaped by "+++", use the AT command as below.

*ATO<CR>*

### 4.2.1 Unicast Data Mode

Unicast data mode can be used for continuous data transmission to a particular node. Unicast data mode can be entered using either ATD or AT+TRANSMITMODE=1 (or 4). If ATD command is used, the device will enter the unicast data mode immediately until escaped using "+++" or reset by ATZ. If the device is reset, it will not enter the unicast data mode in this case.

On the other hand, if AT+TRANSMITMODE=1 (or 4) is used to enter the unicast data mode, the device enters the unicast data mode again whenever the device is reset. To disable the unicast data mode, escape the data mode first by using "+++" string and use AT+TRANSMITMODE=0 and reset the device to apply the setting permanently.

AT commands used for the unicast data mode are as below.

*AT+DESTLA=<value>*
Set the destination address, where <value> is 64-bit IEEE address of destination node for unicast data mode.

*ATD<CR>*

*Set unicast data mode to the preconfigured destination IEEE address without reset.*

*ATD<value><CR>*

Set unicast data mode and <value> can be either 16-bit node ID or 64-bit IEEE address of remote node.

*AT+TRANSMITMODE=1<CR>*

Set the transmit mode to unicast to a destination IEEE address. The device should be reset to apply the mode.

*AT+TRANSMITMODE=4<CR>*

Set the transmit mode to unicast to a sink node. A sink node is determined if s-register 61 is set. The device should be reset to apply the mode.

Example)

```
AT+DESTLA=00019500002FDC40<CR>    # Set the destination IEEE address
OK<CR>                # Response
ATD<CR>               # Enter the unicast mode using destination IEEE address
OK<CR>                # Response
Send unicast msg      # Put the unicast message
+++                   # Leave data mode
OK<CR>                # Response
```

```
ATD0000<CR>           # Enter the unicast mode using 16-bit node ID
OK<CR>                # Response
Send unicast msg      # Put the unicast message
+++                   # Leave data mode
OK<CR>                # Response
```

```
ATD00019500002FDC40<CR>     # Enter the unicast mode using 64-bit IEEE address
OK<CR>                      # Response
Send unicast msg            # Put the unicast message
+++                         # Leave data mode
OK<CR>                      # Response
```

```
AT+DESTLA=00019500002FDC40<CR>    # Set the destination IEEE address
OK<CR>                  # Response
```

```
AT+TRANSMITMODE=1<CR>   # Set transmit mode to destination IEEE address
OK<CR>                  # Response
ATZ<CR>                 # Apply to change
OK<CR>                  # Response
Send unicast msg        # Put the unicast message
+++                     # Leave data mode
OK<CR>                  # Response
```

```
AT+TRANSMITMODE=4<CR>   # Set transmit mode to sink node in the network
OK<CR>                  # Response
ATZ<CR>                 # Apply to change
OK<CR>                  # Response
Send unicast msg        # Put the unicast message
+++                     # Leave data mode
OK<CR>                  # Response
```

#### 4.2.2 Multicast Data Mode

Multicast data mode can be used for continuous data transmission to a particular group of multiple devices. Every ZE10 belongs to a certain group by predetermined group ID when form or join the network.

Multicast data mode can be entered using either ATM or AT+TRANSMITMODE=2. If ATD command is used, the device will enter the multicast data mode immediately until escaped using “+++” or reset by ATZ. If the device is reset, it will not enter the multicast data mode in this case.

On the other hand, if AT+TRANSMITMODE=2 is used to enter the multicast data mode, the device enters the multicast data mode again whenever the device is reset. To disable the multicast data mode, escape the data mode first by using “+++” string and use AT+TRANSMITMODE=0 and reset the device to apply the setting permanently.

AT commands used for the multicast data mode are as below.

*AT+GROUPID=<value>, where <value> is 16-bit group ID for multicast data mode.*
Set Group ID of the device. The device will receive the multicast data sent to this Group ID.

*AT+DESTGROUPID=<value>, where <value> is 16-bit destination group ID for multicast data mode.*
Set destination Group ID of the device.

*ATM<CR>*

*Set multicast data mode to preconfigured destination group ID without reset.*

*ATM<value><CR>*
Set multicast data mode and <value> can be 16-bit group ID.

*AT+TRANSMITMODE=2<CR>*
*Set the transmit mode to multicast mode. The device should be reset to apply the mode.*

*ATS36=n, where n is the radius.*
*ATS37=n, where n is the number of hops through nonmembers.*
Set data relay and forwarding options. Please refer to the ***9 S-Registers*** for more details.

Example)

```
AT+DESTGROUPID=0001<CR>    # Set the destination group ID
OK<CR>                  # Response
ATM<CR>                 # Enter the multicast mode using destination group ID
OK<CR>                  # Response
Send multicast msg      # Put the multicast message
+++                     # Leave data mode
OK<CR>                  # Response
```

```
ATM0001<CR>             # Enter the multicast mode using 16-bit group ID
OK<CR>                  # Response
OK<CR>                  # Response
Send multicast msg      # Put the multicast message
+++                     # Leave data mode
OK<CR>                  # Response
```

```
AT+DESTGROUPID=0001<CR>    # Set the destination group ID
OK<CR>                     # Response
AT+TRANSMITMODE=2<CR>      # Set transmit mode to destination group ID
OK<CR>                     # Response
ATZ<CR>                    # Apply to change
OK<CR>                     # Response
Send multicast msg         # Put the multicast message
+++                        # Leave data mode
OK<CR>                     # Response
```

### 4.2.3 Broadcast Data Mode

Broadcast data mode can be used for continuous data transmission to all devices in the current network. The destination address or group ID of the remote nodes is not necessary for broadcast data mode. The transmit range is determined by the S-register 35.

Broadcast data mode can be entered using either ATB or AT+TRANSMITMODE=3. If ATB command is used, the device will enter the broadcast data mode immediately until escaped using "+++" or reset by ATZ. If the device is reset, it will not enter the broadcast data mode in this case.

On the other hand, if AT+TRANSMITMODE=3 is used to enter the broadcast data mode, the device enters the broadcast data mode again whenever the device is reset. To disable the broadcast data mode, escape the data mode first by using "+++" string and use AT+TRANSMITMODE=0 and reset the device to apply the setting permanently.

AT commands used for the broadcast data mode are as below.

*ATB<CR>*
Set broadcast data mode.

*AT+TRANSMITMODE=3<CR>*
Set the transmit mode to broadcast. It should be reset to apply the mode.

*ATS35=n, where n is the broadcast range.*
*ATS36=n, where n is the radius.*
Set data transmission range. Please refer to the chapter ***9 S-Registers*** for more details.

Example)

```
ATS35=2<CR>             # Set broadcast range to all nodes in the network
OK<CR>                  # Response
ATB<CR>                 # Enter the broadcast mode
OK<CR>                  # Response
Send broadcast msg      # Put the broadcast message
+++                     # Leave data mode
OK<CR>                  # Response
```

```
AT+TRANSMITMODE=3<CR>   # Set transmit mode to current network
OK<CR>                  # Response
```

```
ATZ<CR>                 # Apply to change
OK<CR>                  # Response
Send broadcast msg      # Put the broadcast message
+++                     # Leave data mode
OK<CR>                  # Response
```

# 5 Digital and Analog I/O

## 5.1 GPIO Configuration

The ZE10 module has total 14 I/O pins. Among these pins, I/O pin 6 is assigned to factory reset only and the user can use the rest 13 pins for general purpose inputs and outputs. The I/O pin assignments are shown in Table 5-1. As shown in Table 5-1, some pins provide special functions built into the firmware. Table 5-2 describes the special functions tied to the specific pins.

Table 5-1 Default I/O Pin Assignments (ZE10)

| Pin Number | GPIO Number | Special Functions | AI Enable |
|---|---|---|---|
| 6 | - | Factory reset / Bootloader (**SW_0**) | X |
| 5 | 0 | Permit joining / Wake-up (**SW_1**) | X |
| 3 | 1 | Power LED (**LED_0**) | X |
| 4 | 2 | Status LED (**LED_1**) | X |
| 7 | 3 | | O (AI_0) |
| 8 | 4 | | O (AI_1) |
| 10 | 5 | | O (AI_2) |
| 11 | 6 | | O (AI_3) |
| 24 | 7 | | X |
| 22 | 8 | UART_CTS | X |
| 23 | 9 | UART_RTS | X |
| 19 | 10 | UART_DTR | X |
| 18 | 11 | UART_DSR | X |
| 17 | 12 | | X |

The AT command below can be used to change the GPIO settings.

*AT+GPIO=<values><CR>  or  AT+GPIO<number>=<value><CR>*
*Where, <values> is a number between 0~5 as explained below:*

- *0  Disabled (unmonitored digital input)*
- *1  Digital input, monitored*
- *2  Digital output, default low*
- *3  Digital output, default high*
- *4  Analog input, single ended (only valid for GPIO3~6)*
- *5  Reserved for pin-specific alternate functionalities (See Table 5-2)*

*Table 5-2 Special GPIO Pins*

| Function | GPIO# | Type | Description |
|---|---|---|---|
| Factory_reset / Bootloader | - | DI | This input is dedicated for the factory reset button. If user wants to restore the ZE10 device settings to factory default, this input needs to be pressed for more than two seconds. It will be applied after the input released again. If this input is pressed while the ZE10 starts, the local node enters bootloader menu and it is possible to upload the firmware. |
| Permit_joining / Wake-up | 0 | DI | This input (falling edge, i.e. High to Low change) is used to permit joining or interrupt sleeping. This function is activated when GPIO0 is set to 5. The functions depend on the node type of the ZE10. If the node is set to coordinator or router, this input works to permit other nodes joining for S-register 22 value. If the node is set to sleepy end-device, it is used for wake-up signal when the node is sleeping. |
| Power LED | 1 | DO | If GPIO1 is set to 5, this output turns ON (High) when power is supplied. If the node is set to an end-device, it will blink periodically. Please refer to ***9 S-Registers*** for S14 regarding the management of the LED status. |
| Status LED | 2 | DO | If GPIO2 is set to 5, this output turns ON (High) when the device joins a network, blinks when joining is permitted, and turns OFF (Low) when leaves the network. Please refer to ***9 S-Registers*** for S14 regarding the management of the LED status. |
| UART_CTS | 8 | DI | This input is used for UART CTS if GPIO8 is set to 5. It should be enabled when CTS flow control is used. |
| UART_RTS | 9 | DO | This output is used for UART RTS if GPIO9 is set to 5. It should be enabled when RTS flow control is used. |
| UART_DTR | 10 | DO | This output is used for UART DTR if GPIO10 is set to 5. |
| UATR_DSR | 11 | DI | This input is used for UART DSR if GPIO11 is set to 5. |

## 5.2 I/O Sampling

The I/O status can be monitored and controlled using the local or remote AT commands. Also the ZE10 can be configured to sample the I/O status and send to a remote device periodically or at specific events.

### 5.2.1 Queried Sampling

The I/O status can be monitored and controlled using the AT commands either locally or remotely. To read the digital I/O line values, the AT+DIO command can be used as below. Return value is a combination of 0 (Low) and 1 (High) if the pins are configured as digital I/O lines. If the pins are configured as analog inputs, the corresponding return values are shown as *.

*AT+DIO?<CR> or AT+DIO<number>?<CR>*
*Where <number> is the specific IO number to read ranges 0~12*

AT+DIO command can also be used to set the digital output lines.

*AT+DIO=<value><CR> or AT+DIO<number>=<value><CR>*
*Where <number> is the specific IO number to read ranges 0~12, and the <values> is a combination of 0 or 1, 0 = Low, 1 = High*

The analog input values can be read using the AT+AI command. Return value is a combination of 16-bit hexadecimal integer between 0~2EE0(0~1200mV) or D120-FFFF(-1200~-1mV) which unit value corresponds to 0.1 mv. To convert the hexadecimal value to the actual voltage, please use the formula below. If the pins are configured as digital I/Os, the corresponding return values are shown as *.

*AT+AI?<CR> or AT+AI<number>?<CR>*
*Where <number> is the specific IO number to read ranges 0~2*
*If the Value is between 0 ~ 2EE0, Voltage (mV) = Value * 0.1*
*If the Value is between D120 ~ FFFF, Voltage (mV) = (Value – 65536) * 0.1*

Example)

```
AT+DIO?<CR>
10001**011100    ← DIO 0, 4, 8, 9, 10 are ON. Pins 5, 6 are configured as
analog inputs
OK
AT+DIO0=0<CR>
OK
AT+DIO?<CR>
00001**011100    ← DIO 0, 4, 8, 9, 10 are ON. Pins 5, 6 are configured as
analog inputs
OK
AT+AIO?<CR>
****,****,192A,13E8     ← AI 2, 3 are used. (GPIO5, 6) The others are
configured as DIO.
OK
```

To get/set the I/O values of the remote devices, remote AT commands can be used as below.

*AT+REMOTE=<node ID or IEEE address of a remote node>,<AT command><CR>*

Example)

```
AT+REMOTE=00019500000F3BA1,AT+DIO?<CR>
10001**011100
OK
```

### 5.2.2 Periodic Sampling

The ZE10 can be configured to send its I/O status to a remote device at a periodic rate. To configure this way, please use the AT commands below:

*AT+DESTLA=<IEEE address><CR>*
*Where <IEEE address> is the long address of a remote node to send sampled GPIO values of local node.*

*ATS42=<value><CR>*
*Where <value> is the GPIO sampling period in seconds. If set to 0, GPIO monitoring is disabled.*

*ATS43=<value><CR>*
*Where <value> is the GPIO monitoring node. If set to 0, the monitoring node is selected by AT+DESTLA. If set to is 1, the sink node is set as the monitoring node.*

*ATZ<CR>*
*Reset the device to apply the changes*

Also the ZE10 can be configured to display periodic sampling data coming from the other nodes. To enable this feature, please use the AT command below:

*ATS11=<value><CR>, where <value> determines whether incoming message will be presented or not. If a user wants to display incoming message in command mode, S11 should be enabled.*

If this feature is enabled, incoming message will display in the form of:
++*<IEEE address>*|<GPIO value><CR>.

### 5.2.3 Change Detection Sampling

The ZE10 module can take an I/O sample and transmit it to a remote device immediately whenever a monitored digital I/O pin changes state. It is possible to configure it with periodic sampling simultaneously. To configure this way, please use the AT commands below:

*AT+DESTLA=<IEEE address><CR>*
*Where <IEEE address> is the long address of a remote node to send sampled GPIO values of local node.*

*ATS44=<value><CR>*
*Set GPIO change detection sampling mode. If <value> is set to 0, GPIO sampling is disabled.*

*ATZ<CR>*
*Reset the device to apply the changes*

### 5.2.4 GPIO Remote Control

The ZE10 module supports that a node can change the digital output status of a remote node by switching local digital input pin and the local node should be possible to monitor the GPIO status. To configure this way, please use the AT commands below:

*AT+SOURCELA=<IEEE address><CR>*
*Where <IEEE address> is the long address of a specific source node for changing the status of digital output pin. If this address is set to "FFFFFFFFFFFFFFFF", a local node allows that any node can control the GPIO status.*

*ATS45=<value><CR>*
*Set GPIO remote control mode. If <value> is set to 0, GPIO control is disabled.*

# 6 Working with Development Kit

The ZE10 development kit provides a quick and easy way to be familiar with the ZE10 and test its functionalities before you build your own PCB board to carry the ZE10. It is always recommended to start with the ZE10 development kit before your own design.

## 6.1 Development Kit Contents

The development kit consists of ZE10 modules and development boards to carry them. The development board provides RS232 and USB interface to interface with computers and/or your target devices. It also provides terminals for digital inputs/outputs and analog inputs. It also provides on/off switches and LEDs to simulate and verify digital inputs/outputs. It also has temperature and light sensor so you can test more practical applications. Below is the list of the items included in the package.

- ProBee ZE10 ZigBee OEM Module w/ Chip Antenna (1)
- ProBee ZE10 ZigBee OEM Module w/ U.FL Connector (1)
- ProBee ZE10 ZigBee OEM Module w/ RPSMA Connector (1)
- 3 dBi Dipole Antenna, RP-SMA Plug, Right-hand Thread (2)
- 12cm U.FL to RPSMA Cable, Right-Hand Thread (1)
- Development Boards (3)
- DC Power Adapters (3)
- RS232 Serial Cable (3)
- USB Cable (3)

## 6.2 Development Board Features

The development board is designed to provide convenient access to the ZE10 interface pins. The development board provides external interfaces as below. Also Figure 6-1 shows the layout of those interfaces of the development board.

- RS232 Interface
- USB Interface
- 3 Pre-defined Digital Inputs: Permit Joining, Factory Default Reset, Hardware Reset
- 8 User Configurable Digital Inputs
- 8 Digital Outputs with LEDs (Active High)
- 4 Analog Inputs
- Temperature Sensor
- Light Sensor

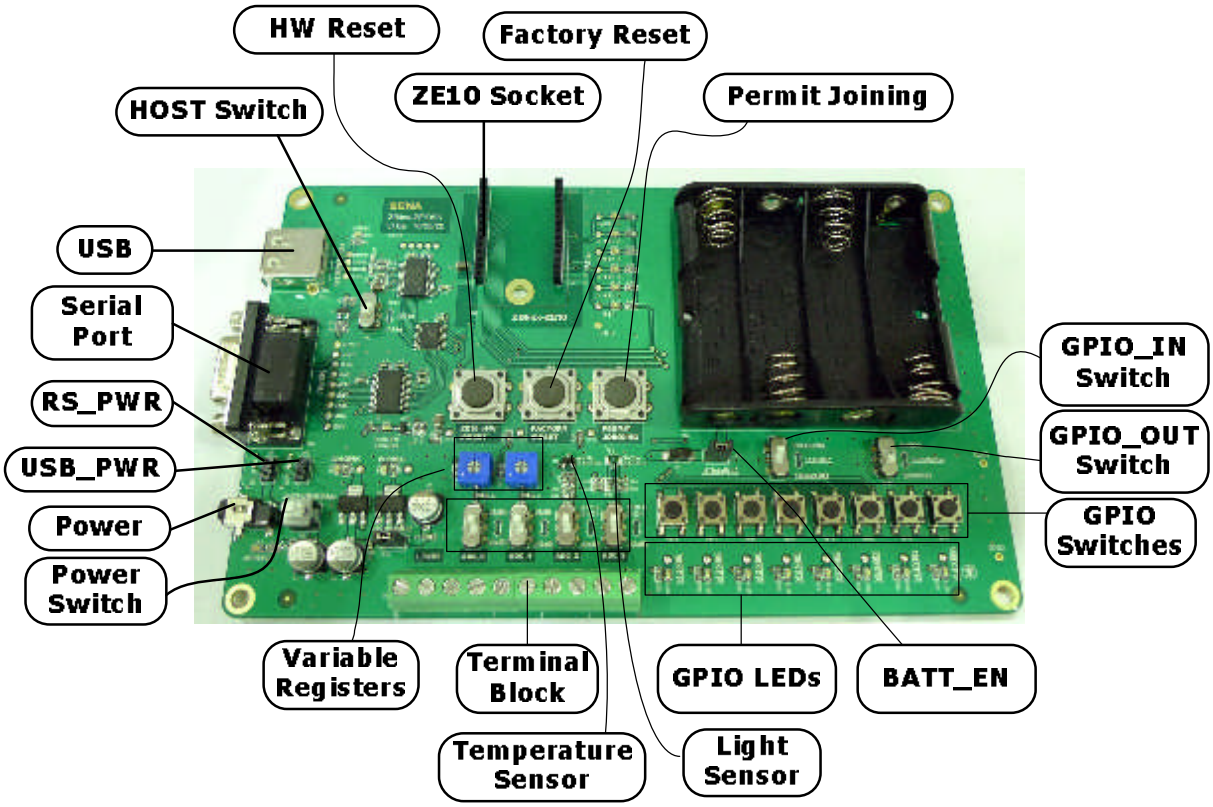



*Figure 6-1 Development Board Layout*

## 6.3 Development Board Hardware Descriptions

### 6.3.1 Power

There are four ways to supply power to the development board:

- 5V Power Port
- USB Port
- AA Batteries
- RS232 Port, 5V through 9th Pin

The RS_PWR, USB_PWR and BATT_EN jumpers MUST be configured properly according to the way. Table 6-1 shows jumpers configuration

*Table 6-1 Jumpers Configuration*

| | RS_PWR | USB_PWR | BATT_EN |
|---|---|---|---|
| **5V Power Port** | Open | Open | Open |
| **USB Port** | Open | Close | Open |
| **AA Batteries** | Open | Open | Close |
| **RS232 Port** | Close | Open | Open |

### 6.3.2 RS232 Port

The development board has an RS232 interface for UART communication. The UART_TXD, UART_RXD, UART_DSR (GPIO_11), UART_DTR (GPIO_10), UART_CTS (GPIO_8) and UART_RTS (GPIO_9) pins of the ZE10 are connected with the RS232 port. To enable the RS232 port, the HOST (SW2) switch should be set to UART.

In addition to the data communication via RS232 port, the RS232 port is capable of supplying power to the development board. In order to use this feature, supply 5V power via the 9$^{th}$ pin of the DB9 socket.

### 6.3.3 USB Port

The development board also has a USB interface for UART communication. To use the USB port, the HOST (SW2) switch should be set to USB and the software driver should be installed on the host computer. The installation CD contains the software.

### 6.3.4 ZE10 GPIO Interface

The development board's external GPIO interfaces as well as functional buttons such as factory default reset switches are internally routed to the ZE10 GPIO pins. Table 6-2 summarizes the development board's interfaces and their corresponding GPIO pin numbers of the ZE10.

*Table 6-2 Development Board GPIO Interface Allocation*

| ZE10 Pin Number | Name | Description |
|---|---|---|
| 3 | GPIO_1 | Digital Output with LED(CR12) Display (Active High)<br>Digital Input with Button Switch (SW10, Active Low or Active High) |
| 4 | GPIO_2 | Digital Output with LED(CR13) Display (Active High)<br>Digital Input with Switch (SW11, Active Low or Active High) |
| 5 | GPIO_0 | Permit Joining Switch (SW9) |
| 6 | Factory Reset | Factory Default Reset Switch (SW8) |
| 7 | GPIO_3 | Digital Output with LED(CR14) Display (Active High)<br>Digital Input with Button Switch (SW12, Active Low or Active High)<br>Analog Input with Variable Register (VR1) |
| 8 | GPIO_4 | Digital Output with LED(CR15) Display (Active High)<br>Digital Input with Button Switch (SW13, Active Low or Active High)<br>Analog Input with Variable Register (VR2) |
| 9 | H/W Reset | Hardware Reset Switch |
| 10 | GPIO_5 | Digital Output with LED(CR16) Display (Active High)<br>Digital Input with Button Switch (SW14, Active Low or Active High)<br>Analog Input with Temperature Sensor (U6) |
| 11 | GPIO_6 | Digital Output with LED(CR17) Display (Active High) |

| | | |
|---|---|---|
| | | Digital Input with Button Switch (SW15, Active Low or Active High)<br>Analog Input with Light Sensor (U7) |
| 17 | GPIO_12 | Digital Output with LED(CR19) Display (Active High)<br>Digital Input with Button Switch (SW17, Active Low or Active High) |
| 18 | GPIO_11 | UART DSR Input |
| 19 | GPIO_10 | UART DTR Output |
| 20 | UART_RXD | UART Data Input |
| 21 | UART_TXD | UART Data Output |
| 22 | GPIO_8 | UART CTS Input |
| 23 | GPIO_9 | UART RTS Output |
| 24 | GPIO_7 | Digital Output with LED(CR18) Display (Active High)<br>Digital Input with Button Switch (SW16, Active Low or Active High) |

### 6.3.5 LEDs (CR12-19)

The development board has 8 LEDs for digital output. To enable the LEDs, the GPIO_OUT (SW20) switch should be set to LED_ON. The LEDs are turned on when the related GPIO pins of the ZE10 are high.

### 6.3.6 Button Switches (SW10-17)

The development board has 8 button switches for digital input. The switches can be configured as active high or active low with the GPIO_IN (SW19) switch.

### 6.3.7 Variable Resistors (VR1, VR2)

There are two variable resistors for analog inputs on the development board. The analog inputs can be read with the GPIO_3 and GPIO_4. To enable the variable resistors, the ADC_0 and ADC_1 switches should be placed to VR_0 and VR_1.

### 6.3.8 Temperature Sensor

The GPIO_5 of the ZE10 is routed to the temperature sensor. The relationship between temperature and analog input is as follows:

Temperature(℃) ={ Volt(mV) -600(mV)} * 0.1(℃/mV)

### 6.3.9 Light Sensor

The GPIO_6 of the ZE10 is routed to the light sensor. Approximately, the relationship between illumination and analog input is as follows:

Illumination(Lux) = Volt(mV) * 0.25(Lux/mV)

### 6.3.10 Terminal Block

The development board has a terminal block for external analog inputs. Figure 6-2 shows the layout of the terminal block. To use external analog inputs instead of the variable resistors, temperature sensor or light sensor, the ADC_0 (SW4), ADC_1 (SW5), ADC_2 (SW6) or ADC_3 (SW7) should be placed on EXT_0, EXT_1, EXT_2, or EXT_3.

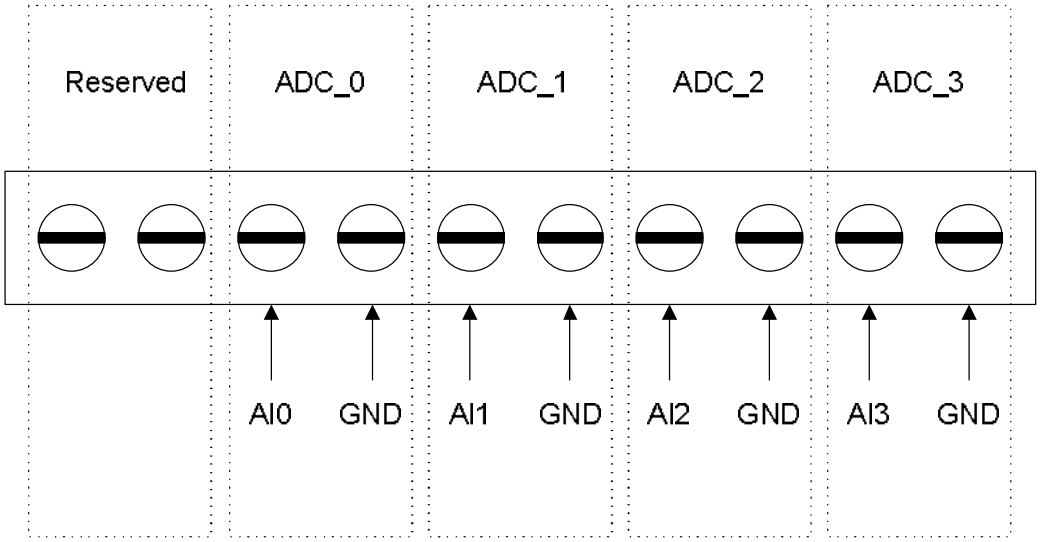



*Figure 6-2 Terminal Block for External Anlalog Inputs*

## 6.4 Configuration Example #1: 1 Coordinator, 1 Router and 1 End-Device

Here is the example configuration to help to start with the ZE10 and the development kit. The goal of this example is to provide hardware configuration and AT commands for the scenario as below:

- One ZE10 module acts as a coordinator and it is connected to a host computer via USB cable.
- One ZE10 module acts as a router and it sends temperature sensor, light sensor and digital inputs to the coordinator every second.
- One ZE10 module acts as a sleepy end device and it sends temperature sensor, light sensor and digital inputs to the coordinator every second.

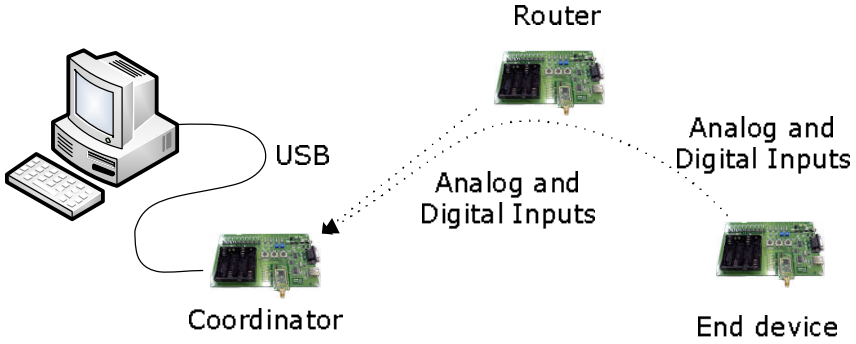



*Figure 6-3 Example #1: 1 Coordinator, 1 Router and 1 End Device*

### 6.4.1 Coordinator Configuration

Hardware configuration for the coordinator is as follows:

- Set the HOST switch to USB
- Close the USB_PWR jumper and open the RS_PWR and BATT_PWR jumpers.
- Connect a USB cable between a host computer and the coordinator

The AT commands for the coordinator are as follows:

```
AT+NODETYPE=1
OK
AT+PANID=7772
OK
ATS11=1
OK
AT+LONGADDR
0001950000000001
OK
ATZ
OK
```

### 6.4.2 Router Configuration

Hardware configuration for the coordinator is as follows:

- Set the ADC_2 and ADC_3 switch to TEMP_SENS and LIGHT_SENS

The AT commands for the router are as follows:

```
AT+NODETYPE=2
OK
AT+PANID=7772
OK
ATS42=1
OK
AT+DESTLA=0001950000000001
OK
AT+GPIO=1111144111111
OK
ATZ
OK
```

### 6.4.3 End-device Configuration

Hardware configuration for the end device is as follows:

- Set the ADC_2 and ADC_3 switch to TEMP_SENS and LIGHT_SENS

The AT commands for coordinator configuration is as follows:

```
AT+NODETYPE=4
OK
AT+PANID=7772
OK
ATS42=1
OK
AT+DESTLA=0001950000000001
OK
AT+GPIO=1111144111111
OK
ATZ
OK
```

### 6.4.4 Monitoring the Sensors

Once the configurations are done and the system is running, the terminal (computer) connected to the coordinator will print out the incoming data as follows:

```
++0001950000000002|10000**000000|****,****,233E,006A
++0001950000000003|10000**000000|****,****,2254,0B23
++0001950000000002|10000**000000|****,****,2235,004E
++0001950000000003|10000**000000|****,****,226C,0B8B
```

## 6.5 Configuration Example #2: 1 Coordinator and 2 End-Devices

Here is another example assuming different system configurations. This example can be described as follows:

- One ZE10 module acts as a coordinator and it is connected to a host computer via serial cable.
- Two ZE10 modules act as end devices.
- Turn on/off the LEDs of the end devices remotely through the coordinator

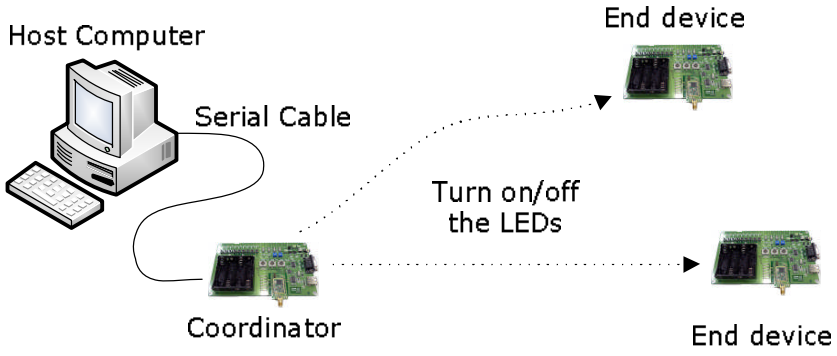



*Figure 6-4 Example #2: 1 Coordinator and 2 End Devices*

### 6.5.1 Coordinator Configuration

Hardware configuration for the coordinator is as follows:

- Set the HOST switch to UART
- Connect a serial cable between a host computer and the coordinator

The AT commands for coordinator configuration is as follows:

```
AT+NODETYPE=1
OK
AT+PANID=7772
OK
ATZ
OK
```

### 6.5.2 End-devices Configuration

Hardware configuration for the coordinator is as follows:

- Set the GPIO_OUT switch to LED_ON

The AT commands for the end devices are as follows:

```
AT+NODETYPE=4
OK
AT+PANID=7772
OK
AT+GPIO=3333333333333
OK
```

```
AT+LONGADDR
0001950000000003
OK
ATZ
OK
```

### 6.5.3 Turning on/off the LEDs

The AT commands to control the LEDs are as follows:

```
AT+REMOTE=0001950000000003,AT+DIO=0000000000000
OK
AT+REMOTE=0001950000000003,AT+DIO=1111111111111
OK
```

# 7 AT Command Reference

## 7.1 Command Line Format

Each command line is composed of a prefix, body and terminator. Only “AT” can be used for the prefix. The body is a string of characters. The terminator is <CR> or <NL>. A command can be one of the followings:

- Read a configuration parameter:

AT<Command>?<CR>

- Do something:

AT<Command><CR>

- Write a configuration parameter:

AT<Command>=<Parameter 1>,<Parameter 2>,<Parameter 3>,…<CR>

Commands denoted with a “+” character are extended AT commands.
S-registers are supported. Multiple commands in the same command line are not supported.

## 7.2 Standard AT Commands

### 7.2.1 AT

| **Description** | Check the state of the connection between host and ProBee device. |
|---|---|
| **Execute** | AT<CR> |
| **Response** | OK<CR> |

### 7.2.2 ATB

| Description | Enter the broadcast data mode. Every data is packetized at intervals of inter character timeout (S12) and transmit range is determined by broadcast range(S35). |
|---|---|
| Execute | ATB<CR> |
| Response | OK<CR> |

### 7.2.3 ATD

| Description | Enter the unicast data mode to specific node. If there’s no argument after the command, it will connect to the preconfigured destination IEEE address. Successful transmission guarantees successful reception and every data is packetized at intervals of inter character timeout(S12). | | |
|---|---|---|---|
| Execute | ATD<CR> | ATD<nodeID><CR> | ATD<IEEE><CR> |
| Response | OK<CR> or ERROR<CR> | | |

#### 7.2.4 ATE

| | |
|---|---|
| Description | Echo(1) or No Echo(0). |
| Execute | ATE<number><CR> |
| Response | OK<CR> |

#### 7.2.5 ATM

| | | |
|---|---|---|
| Description | Enter the multicast mode. If there's no argument after the command, it will connect to the preconfigured destination group ID. Every data is packetized at intervals of inter character timeout(S12). | |
| Execute | ATM<CR> | ATM<groupID><CR> |
| Response | OK<CR> | |

#### 7.2.6 ATO

| | |
|---|---|
| Description | Return to data mode. |
| Execute | ATO<CR> |
| Response | |

#### 7.2.7 ATR

| | |
|---|---|
| Description | Response(1) or No Response(0). |
| Execute | ATR<number><CR> |
| Response | OK<CR> |

#### 7.2.8 ATS

| | | | |
|---|---|---|---|
| Description | Get/Set the S-registers. | | |
| Execute | ATS<CR> | ATS<number>?<CR> | ATS<number>=<value><CR> |
| Response | List of the S-register and description | <value><CR><br>OK<CR> | OK<CR> or ERROR<CR> |

#### 7.2.9 ATZ

| | |
|---|---|
| Description | Reset the node. |
| Execute | ATZ<CR> |
| Response | OK<CR> |

#### 7.2.10 AT&F

| | |
|---|---|
| Description | Factory reset. |
| Execute | AT&F<CR> |
| Response | OK<CR> |

#### 7.2.11 +++

| | |
|---|---|
| Description | Interrupt data mode. “+++” must be entered within 500ms. |
| Execute | +++ |
| Response | OK<CR> |

## 7.3 Node Specific Information

### 7.3.1 AT+LONGADDR or AT+LA

| | |
|---|---|
| Description | Get the 64-bit IEEE address. (EUI address, long address) The IEEE address is a unique device address assigned during manufacturing. This address is unique to each physical device. |
| Execute | AT+LONGADDR?<CR> |
| Response | <IEEE address><CR><br>OK<CR> |

### 7.3.2 AT+SHORTADDR or AT+SA

| | |
|---|---|
| Description | Get the 16-bit Node ID. (network address, short address) The node ID of “0000” is reserved for the coordinator. All other devices receive a randomly generated address from the router or coordinator device that allows the join. This value can change if an address conflict is detected where two devices are found to have the same address or a device leaves the network and later joins. |
| Execute | AT+SHORTADDR?<CR> |
| Response | <node ID><CR><br>OK<CR> |

### 7.3.3 AT+ADDRDISCOVERY or AT+AD

| | | |
|---|---|---|
| Description | Get the corresponding address of a node whose long or short address is known. | |
| Execute | AT+ADDRDISCOVERY=<IEEE><CR> | AT+ADDRDISCOVERY=<nodeID><CR> |
| Response | <node ID><CR><br>OK<CR> | <IEEE address><CR><br>OK<CR> |

### 7.3.4 AT+OPCH or AT+OC

| | |
|---|---|
| Description | Get the operating channel number used for transmitting and receiving between RF modules. Uses 802.15.4 channel numbers. A value of “0” means the device has not joined a PAN and is not operating on any channel. |
| Execute | AT+OPCH?<CR> |
| Response | <value><CR><br>OK<CR> |

#### 7.3.5 AT+OPPANID or AT+OI

| | |
|---|---|
| Description | Get the operating 16-bit PAN ID. If PAN ID is set specific value not 0s, operating PAN ID will equal designated PAN ID. A value of “FFFF” means the device is not joined a network. |
| Execute | AT+OPPANID?<CR> |
| Response | <value><CR><br>OK<CR> |

#### 7.3.6 AT+OPEPID or AT+OE

| | |
|---|---|
| Description | Get the operating 64-bit extended PAN ID. A value of “FFFFFFFFFFFFFFFF” means the device is not joined a network. |
| Execute | AT+OPEPID?<CR> |
| Response | <value><CR><br>OK<CR> |

#### 7.3.7 AT+VERSION or AT+VR

| | |
|---|---|
| Description | Get firmware version of the module. The firmware version returns a form of “PTvx.x”. |
| Execute | AT+VERSION?<CR> |
| Response | <value><CR><br>OK<CR> |

#### 7.3.8 AT+PRODUCTNAME or AT+PN

| | |
|---|---|
| Description | Get the product name. It can be used to distinguish among different hardware platforms. |
| Execute | AT+PRODUCTNAME?<CR> |
| Response | <value><CR><br>OK<CR> |

### 7.4 Network Specific Parameters

#### 7.4.1 AT+CHMASK or AT+CM

| | | |
|---|---|---|
| Description | Get/Set the channel mask to use. Bits set as 1 indicate that this particular channel can be used. Bits set as 0 indicate that this particular channel should not be used. For example, a channel mask value of 0x00000001 indicates that only channel 0 should be used. Valid channels range from 12 to 25 inclusive. (14 channels are available.) This translates to a channel mask value of “0x03FFF000”. | |
| Execute | AT+CHMASK?<CR> | AT+CHMASK=<value><CR> |
| Response | <value><CR><br>OK<CR> | OK<CR> or ERROR<CR> |
| Default | 03FFF000 | |

| | |
|---|---|
| Range | 00001000 - 03FFF000 |

### 7.4.2 AT+PANID or AT+PI

| | | |
|---|---|---|
| Description | Get/Set the 16-bit PAN ID. If it is set to 0s, the coordinator will select a random 16-bit hexadecimal number as PAN ID, and the router/end-device will join any existing network if it is allowed. | |
| Execute | AT+PANID?<CR> | AT+PANID=<value><CR> |
| Response | <value><CR><br>OK<CR> | OK<CR> or ERROR<CR> |
| Default | 0000 | |
| Range | 0000 – FFFF | |

### 7.4.3 AT+EPID or AT+EI

| | | |
|---|---|---|
| Description | Get/Set the 64-bit extended PAN ID. If the node is a coordinator and the EPID is set to 0s, it will take the rearranged IEEE address as random number. | |
| Execute | AT+EPID?<CR> | AT+EPID=<value><CR> |
| Response | <value><CR><br>OK<CR> | OK<CR> or ERROR<CR> |
| Default | 0000000000000000 | |
| Range | 0000000000000000 - FFFFFFFFFFFFFFFF | |

### 7.4.4 AT+GROUPID or AT+GI

| | | |
|---|---|---|
| Description | Get/Set the 16-bit Group ID. It is used for multicast data reception. | |
| Execute | AT+GROUPID?<CR> | AT+GROUPID=<value><CR> |
| Response | <value><CR><br>OK<CR> | OK<CR> or ERROR<CR> |
| Default | 0000 | |
| Range | 0000 - FFFF | |

### 7.4.5 AT+POWER or AT+PW

| | | |
|---|---|---|
| Description | Get/Set the TX power at which the RF module transmits conducted power. | |
| Execute | AT+POWER?<CR> | AT+POWER=<value><CR> |
| Response | <value><CR><br>OK<CR> | OK<CR> or ERROR<CR> |
| Default | 3 | |
| Range | -43 - 3 | |

### 7.4.6 AT+STACK or AT+ZS

| | |
|---|---|
| Description | Get/Set the ZigBee stack profile value. This value must be set the same on all devices |

| | | |
|---|---|---|
| | that should join the same network. (0: Network specific, 1: ZigBee, 2: ZigBee Pro) | |
| Execute | AT+STACK?<CR> | AT+STACK=<value><CR> |
| Response | <value><CR><br>OK<CR> | OK<CR> or ERROR<CR> |
| Default | 0 | |
| Range | 0 - 2 | |

#### 7.4.7 AT+NODENAME or AT+NN

| | | |
|---|---|---|
| Description | Get/Set the node name. It can be used to make a clear distinction role or functionality of local node from others. | |
| Execute | AT+NODENAME?<CR> | AT+NODENAME=<value><CR> |
| Response | <value><CR><br>OK<CR> | OK<CR> or ERROR<CR> |
| Default | ProBee-ZE | |
| Range | Up to 16 characters | |

### 7.5 Network Formation and Join

#### 7.5.1 AT+ASCAN or AT+AS

| | | |
|---|---|---|
| Description | Perform a scanning for existing network. It can take up to 5 seconds.<br>(CAUTION: It is possible to cause a link error if a node is a parent of an end-device.) | |
| Execute | AT+ASCAN<CR> | AT+ASCAN=<channel mask><CR> |
| Response | <channel>\|<PAN ID>\|<EPID>\|<joinable>\|<stack profile><br>*List of the existing networks …*<br>OK<CR> | |

#### 7.5.2 AT+ESCAN or AT+ES

| | | |
|---|---|---|
| Description | Perform a scanning for energy level of all channels. It can take up to 5 seconds.<br>(CAUTION: It is possible to cause a link error if a node is a parent of an end-device.) | |
| Execute | AT+ESCAN<CR> | AT+ESCAN=<channel mask><CR> |
| Response | <channel>\|<RSSI ><br>*List of the RSSI for each channel …*<br>OK<CR> | |

#### 7.5.3 AT+DSCAN or AT+DS

| | |
|---|---|
| Description | Perform a scanning for existing nodes in the current network. A coordinator or router can have children nodes and they are presented by “CLDn”. It can be possible to place some conditions such as scan duration, node type and node name in order. The information of local node is presented by “*”. |
| Execute | AT+DSCAN<CR> or |

| | |
|---|---|
| | AT+DSCAN=<scan duration><CR> or<br>AT+DSCAN=<scan duration>,<node type><CR> or<br>AT+DSCAN=<scan duration>,<node type>,<node name><CR> |
| Response | <node type>\|<IEEE address>\|<node ID>\|<version>\|<product name>\|<node name><br>*List of the existing nodes and information…*<br>OK<CR> |
| Range | Scan duration: 1 – 10<br>Node type: 0 – 4 (Zero indicates all node types.) |

#### 7.5.4 AT+NODETYPE or AT+NT

| | | |
|---|---|---|
| Description | Get/Set the node type. (0: none, 1: coordinator, 2: router, 3: end-device, 4: sleepy end-device) | |
| Execute | AT+NODETYPE?<CR> | AT+NODETYPE=<value><CR> |
| Response | <value><CR><br>OK<CR> | OK<CR> or ERROR<CR> |
| Default | 0 | |
| Range | 0 - 4 | |

#### 7.5.5 AT+PERMIT or AT+PJ

| | | |
|---|---|---|
| Description | Permit other nodes to join the network. (0~254s, 255=continuance) An end-device cannot perform it. | |
| Execute | AT+PERMIT<CR> | AT+PERMIT=<value><CR> |
| Response | OK<CR> | |
| Range | 0 - 255 | |

#### 7.5.6 AT+LEAVE or AT+LV

| | |
|---|---|
| Description | Leave the current network. |
| Execute | AT+LEAVE<CR> |
| Response | OK<CR> |

### 7.6 End-device

#### 7.6.1 AT+SLEEP or AT+SM

| | |
|---|---|
| Description | Enter the sleep mode immediately if the node is an end-device. |
| Execute | AT+SLEEP<CR> |
| Response | OK<CR> |

#### 7.6.2 AT+PARENTLA or AT+PL

| | |
|---|---|
| Description | Get the 64-bit IEEE address of the parent node. |
| Execute | AT+PARENTLA?<CR> |

| | |
|---|---|
| Response | <IEEE address><CR><br>OK<CR> |

#### 7.6.3 AT+PARENTSA or AT+PS

| | |
|---|---|
| Description | Get the 16-bit Node ID of the parent node. |
| Execute | AT+PARENTSA?<CR> |
| Response | <node ID><CR><br>OK<CR> |

#### 7.6.4 AT+CHILDTABLE or AT+CT

| | |
|---|---|
| Description | Get the child table composed of the end-devices. |
| Execute | AT+CHILDTABLE?<CR> |
| Response | <node type>\|<IEEE address>\|<node ID><br>*List of the child nodes…*<br>OK<CR> |

### 7.7 Data Transmission and Reception

#### 7.7.1 AT+DESTLA or AT+DL

| | | |
|---|---|---|
| Description | Get/Set the 64-bit destination IEEE address for unicast data transmission. | |
| Execute | AT+DESTLA?<CR> | AT+DESTLA=<value><CR> |
| Response | <value><CR><br>OK<CR> | OK<CR> or ERROR<CR> |
| Default | 0000000000000000 | |
| Range | 0000000000000000 - FFFFFFFFFFFFFFFF | |

#### 7.7.2 AT+DESTGROUPID or AT+DG

| | | |
|---|---|---|
| Description | Get/Set the 16-bit destination Group ID for multicast data transmission. | |
| Execute | AT+DESTGROUPID?<CR> | AT+ DESTGROUPID=<value><CR> |
| Response | <value><CR><br>OK<CR> | OK<CR> or ERROR<CR> |
| Default | 0000 | |
| Range | 0000 – FFFF | |

#### 7.7.3 AT+SINKINFO or AT+SI

| | |
|---|---|
| Description | Get the long and short addresses of sink node. If there is no sink node in the network, the addresses will be “FFFFFFFFFFFFFFFF” and “FFFF”. |
| Execute | AT+SINKINFO?<CR> |
| Response | <IEEE address>,<nodeID><CR><br>OK<CR> |

#### 7.7.4 AT+MAXPAYLOAD or AT+MP

| | |
|---|---|
| Description | Get the maximum size of the payload that the Application Support sub-layer will accept. The size depends on the security level in use. |
| Execute | AT+MAXPAYLOAD?<CR> |
| Response | <value><CR><br>OK<CR> |

#### 7.7.5 AT+TRASNSMITMODE or AT+TM

| | | |
|---|---|---|
| Description | Get/Set the data transmission mode. This mode is maintained after rebooting. (0: none, 1: unicast to destination long address, 2: multicast to destination group ID, 3: broadcast to all nodes, 4: unicast to sink node) | |
| Execute | AT+TRANSMITMODE?<CR> | AT+TRANSMITMODE=<value><CR> |
| Response | <value><CR><br>OK<CR> | OK<CR> or ERROR<CR> |
| Default | 0 | |
| Range | 0 - 4 | |

#### 7.7.6 AT+UNICAST or AT+UC

| | | |
|---|---|---|
| Description | Send unicast message to specific node using long or short address of remote node. Up to 90 bytes (or 72 bytes when security enabled) are sent to the node at once. | |
| Execute | AT+UNICAST=<IEEE>,<data><CR> | AT+UNICAST=<nodeID>,<data><CR> |
| Response | OK<CR> | |

#### 7.7.7 AT+MULTICAST or AT+MC

| | |
|---|---|
| Description | Send multicast message to specific group. Every node belongs to a group after being booted up if it is designated a group ID. |
| Execute | AT+MULTICAST=<groupID>,<data><CR> |
| Response | OK<CR> |

#### 7.7.8 AT+BROADCAST or AT+BC

| | |
|---|---|
| Description | Send broadcast message to the all nodes joining the network. Transmit range is determined by broadcast range (S35). |
| Execute | AT+BROADCAST=<data><CR> |
| Response | OK<CR> |

#### 7.7.9 AT+REMOTE or AT+RC

| | |
|---|---|
| Description | Get/Set the configuration values of remote node using long or short address. AT+REMOTE<CR> shows the list of possible commands. It is possible to check the |

| | | |
|---|---|---|
| | available remote command by “AT+REMOTE<CR>”. | |
| Execute | AT+REMOTE=<addr>,<cmd>?<CR> | AT+REMOTE=<addr>,<cmd>=<value><CR> |
| Response | <value><CR><br>OK<CR> | OK<CR> |

## 7.8 UART

### 7.8.1 AT+BAUDRATE or AT+UB

| | | |
|---|---|---|
| Description | Get/Set the serial port speed for communication between the module serial port and host. | |
| Execute | AT+BAUDRATE?<CR> | AT+BAUDRATE=<value><CR> |
| Response | <value><CR><br>OK<CR> | OK<CR> or ERROR<CR> |
| Default | 9600 | |
| Range | 300 - 230400 | |

### 7.8.2 AT+DATABIT or AT+UD

| | | |
|---|---|---|
| Description | Get/Set the serial data bit setting on the module. | |
| Execute | AT+DATABIT?<CR> | AT+DATABIT=<value><CR> |
| Response | <value><CR><br>OK<CR> | OK<CR> or ERROR<CR> |
| Default | 8 | |
| Range | 7 - 8 | |

### 7.8.3 AT+PARITY or AT+UP

| | | |
|---|---|---|
| Description | Get/Set the serial parity setting on the module. | |
| Execute | AT+PARITY?<CR> | AT+PARITY=<value><CR> |
| Response | <value><CR><br>OK<CR> | OK<CR> or ERROR<CR> |
| Default | 0 | |
| Range | 0 - 2 | |

### 7.8.4 AT+STOPBIT or AT+US

| | | |
|---|---|---|
| Description | Get/Set the serial stop bit setting on the module. | |
| Execute | AT+STOPBIT?<CR> | AT+STOPBIT=<value><CR> |
| Response | <value><CR><br>OK<CR> | OK<CR> or ERROR<CR> |
| Default | 1 | |
| Range | 1 - 2 | |

#### 7.8.5 AT+FLOWCTRL or AT+UF

| | | |
|---|---|---|
| Description | Get/Set flow control status. (0: none, 1: software, 2: hardware flow control) | |
| Execute | AT+FLOWCTRL?<CR> | AT+FLOWCTRL=<value><CR> |
| Response | <value><CR><br>OK<CR> | OK<CR> or ERROR<CR> |
| Default | 2 | |
| Range | 0 - 2 | |

## 7.9 Security

#### 7.9.1 AT+SECURITY or AT+SE

| | | |
|---|---|---|
| Description | Get/Set the security status. (0: turning security off, 1: on) Every node must be under the same security level in the network. | |
| Execute | AT+SECURITY?<CR> | AT+SECURITY=<value><CR> |
| Response | <value><CR><br>OK<CR> | OK<CR> or ERROR<CR> |
| Default | 0 | |
| Range | 0 - 1 | |

#### 7.9.2 AT+LINKKEY or AT+LK

| | | |
|---|---|---|
| Description | Get/Set the 128-bit link key. (0 or 0s: generate 128-bit random number) | |
| Execute | AT+LINKKEY?<CR> | AT+LINKKEY=<value><CR> |
| Response | <value><CR><br>OK<CR> | OK<CR> or ERROR<CR> |
| Default | FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF | |

#### 7.9.3 AT+NWKKEY or AT+NK

| | | |
|---|---|---|
| Description | Get/Set the 128-bit network key. (0 or 0s: generate random 128-bit number) | |
| Execute | AT+NWKKEY?<CR> | AT+NWKKEY=<value><CR> |
| Response | <value><CR><br>OK<CR> | OK<CR> or ERROR<CR> |
| Default | FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF | |

#### 7.9.4 AT+UPDATEKEY or AT+UK

| | |
|---|---|
| Description | Update the network key with a new generated key. (possible only for coordinator) It is applied after 30s later because sending too quickly may cause a sleepy end-device to miss the alternate encryption key and only get the switch key message, which means it will be unable to change to the new network key. |
| Execute | AT+UPDATEKEY<CR> |

| | |
|---|---|
| Response | OK<CR> or ERROR<CR> |

## 7.10 GPIO

### 7.10.1 AT+GPIO or AT+IO

| | | |
|---|---|---|
| Description | Get/Set the configuration status of GPIO individually or all at once. Only GPIO3-6 can be set 4, others are available 0-3 and 5. | |
| Execute | AT+GPIO?<CR> or<br>AT+GPIO<number>?<CR> | AT+GPIO=<values><CR> or<br>AT+GPIO<number>=<value><CR> |
| Response | <value><CR><br>OK<CR> | OK<CR> or ERROR<CR> |
| Default | 5550000055550 | |
| Range | GPIO3-6: 0-5, the others: 0-3 and 5<br>0 Disabled (unmonitored digital input)<br>1 Digital input, monitored<br>2 Digital output, default low<br>3 Digital output, default high<br>4 Analog input, single ended (only valid for GPIO3~6)<br>5 Reserved for pin-specific alternate functionalities (See Table 5-2) | |

### 7.10.2 AT+DIO

| | | |
|---|---|---|
| Description | Get/Set the digital IO value individually or all at once. All GPIO pins can be used as digital IO and it is only possible to set 0 or 1. If there are some pins used analog input, they are displayed by "*". | |
| Execute | AT+DIO?<CR> or<br>AT+DIO<number>?<CR> | AT+DIO=<values><CR> or<br>AT+DIO<number>=<value><CR> |
| Response | <value><CR><br>OK<CR> | OK<CR> or ERROR<CR> |
| Default | 0000000000000 | |
| Range | 0000000000000 - 1111111111111 | |

### 7.10.3 AT+AI

| | | |
|---|---|---|
| Description | Get the analog input value individually or all at once. GPIO3~6 are possible to set ADC and it's read only. If there are some pins used digital IO, they are displayed by "****". | |
| Execute | AT+AI?<CR> | AT+AI<number>?<CR> |
| Response | <value>,<value>,<value>,<value><CR><br>OK<CR> | <value><CR><br>OK<CR> |

### 7.10.4 AT+ SOURCELA or AT+SL

| | |
|---|---|
| Description | Get/Set the 64-bit source IEEE address for GPIO remote control. It is possible to limit |

| | | |
|---|---|---|
| | on the implementation of this functionality by setting specific address. If set to default value, "FFFFFFFFFFFFFFFF", the node allows that any node can control the GPIO status. | |
| Execute | AT+SOURCELA?<CR> | AT+SOURCELA=<value><CR> |
| Response | <value><CR><br>OK<CR> | OK<CR> or ERROR<CR> |
| Default | FFFFFFFFFFFFFFFF | |
| Range | 0000000000000000 – FFFFFFFFFFFFFFFF | |

## 7.11 Firmware Upload and Help

### 7.11.1 AT+BOOTLOAD or AT+BL

| | | |
|---|---|---|
| Description | Start bootloader to upload firmware. (blank: local, 1: passthru, 2: clone)<br>The device leaves the AT command line and enters the bootloader menu for uploading new ProBee firmware. There's no restrict to use local upload but a sleepy end device cannot upload to remote node over the air. The current UART should be set 115200bps, 8 data bits, no parity, 1 stop bit and none flow control. | |
| Execute | AT+BOOTLOAD<CR> | AT+BOOTLOAD=<number>,<IEEE><CR> |
| Response | *Bootloader menu*<br>…<br>Serial upload complete | Please start .ebl upload image ...<CR> or<br>Start cloning ...<CR><br>*Bootloader menu*<br>…<br>Bootload Complete! |

### 7.11.2 AT+HELP or AT+HP

| | |
|---|---|
| Description | Show all the available commands |
| Execute | AT+HELP<CR> |
| Response | Available AT Commands<br>…<br>OK<CR> |

# 8 AT Command Examples

## 8.1 Network configuration

### 8.1.1 Coordinator

```
AT+NODETYPE=1<CR>              # set node type (coordinator)
OK                             # Response
ATZ<CR>                        # Apply the change
OK                             # Response
```

### 8.1.2 Router

```
AT+NODETYPE=2<CR>              # set node type (router)
OK                             # Response
ATZ<CR>                        # Apply the change
OK                             # Response
```

### 8.1.3 End-device

```
AT+NODETYPE=3<CR>              # set node type (end-device)
OK                             # Response
ATZ<CR>                        # Apply the change
OK                             # Response
```

### 8.1.4 Sleepy End-device

```
AT+NODETYPE=4<CR>              # set node type (sleepy end-device)
OK                             # Response
ATZ<CR>                        # Apply the change
OK                             # Response
```

### 8.1.5 Specifying Network

```
AT+CHMASK=1000<CR>             # set 32-bit channel mask (ch: 12)
OK                             # Response
AT+PANID=0001<CR>              # set 16-bit PAN ID (0001)
OK                             # Response
AT+EPID=0000000000000001<CR>   # set 64-bit extended PAN ID (0000000000000001)
OK                             # Response
AT+ADDRDISCOVERY=1FEF<CR>  # get 64-bit address using 16-bit short address
```

```
00019500002FDC15
OK                          # Response
AT+REMOTE=1FEF,AT+LONGADDR<CR>    # get 64-bit address of remote node (1FEF)
00019500002FDC15
OK                          # Response
AT+DSCAN<CR>                # get the information of other nodes in the network
ZC*|00019500002FDC40|0000|PTv1.0|ZE10|ZE10_COM3
ZR |00019500002FDC15|1FEF|PTv1.0|ZE10|ZE10_COM4
CLD0|00019500002FDC17
CLD1|00019500002FDC14
ZED|00019500002FDC17|7E34|PTv1.0|ZS10|ZS10_COM5
SED|00019500002FDC14|F5AC|PTv1.0|ZU10|ZU10_COM6
OK                          # Response
```

## 8.2 Data Transmission

### 8.2.1 Command Mode

```
# send unicast message(MSG01) to specific node using 64-bit long
address(0x00019500002FDC15)
AT+UNICAST=00019500002FDC15,MSG01<CR>
OK                              # Response
# send unicast message(MSG02) to specific node using 16-bit short
address(0x1FEF)
AT+UNICAST=1FEF,MSG02<CR>
OK                              # Response
# send multicast message(MSG03) to specific group(0001)
AT+MULTICAST=0001,MSG03<CR>
OK                              # Response
# send broadcast message(MSG04) to current network and the range is
determined by S35.
AT+BROADCAST=MSG04<CR>
OK                              # Response
```

### 8.2.2 Data Mode

```
ATD<CR>              # enter the unicast mode using destination long address
OK                   # Response
MSG05+++             # send data(MSG05) and leave data mode
ATD0000<CR>             # enter the unicast mode using 16-bit short address
```

```
OK                          # Response
MSG06+++                    # send data(MSG06) and leave data mode
ATD00019500002FDC40<CR>     # enter the unicast mode using 64-bit long address
OK                          # Response
MSG07+++                    # send data(MSG07) and leave data mode
ATM<CR>                     # enter the multicast mode using destination group ID
OK                          # Response
MSG08+++                    # send data(MSG08) and leave data mode
ATM0001<CR>                 # enter the multicast mode using 16-bit group ID
OK                          # Response
MSG09+++                    # send data(MSG09) and leave data mode
ATB<CR>                     # enter the broadcast mode
OK                          # Response
MSG10+++                    # send data(MSG10) and leave data mode
AT+TRANSMITMODE=1<CR>       # set transmit mode
OK                          # Response
ATZ<CR>                     # Apply the change
OK                          # Response
MSG11+++                    # send data(MSG11) and leave data mode
ATO<CR>                     # return to the current data mode
```

## 8.3 UART Settings

```
AT+BAUDRATE=115200<CR>      # set baud rate (115,200bps)
OK                          # Response
AT+DATABIT=8<CR>            # set data bit (8bit)
OK                          # Response
AT+PARITY=0<CR>             # set parity (none)
OK                          # Response
AT+STOPBIT=1<CR>            # set stop bit (1bit)
OK                          # Response
AT+FLOWCTRL=2<CR>           # set flow control (hardware)
OK                          # Response
ATZ<CR>                     # Apply the change
OK                          # Response
```

## 8.4 Security

```
AT+SECURITY=1<CR>          # set security level (ON)
OK                         # Response
AT+LINKKEY=5A46…21<CR>     # set the 128-bit trust center link key (5A46…21)
ATZ<CR>                    # Apply the change
OK                         # Response
AT+NWKKEY=0<CR>            # set the 128-bit network key (random key)
OK                         # Response
AT+UPDATEKEY<CR>           # update the changed network key
OK                         # Response
```

## 8.5 GPIO

```
AT+GPIO=5500044112233<CR>  # set GPIO configuration
OK                         # Response
ATZ<CR>                    # Apply the change
OK                         # Response
AT+GPIO0<CR>               # check the current GPIO status (GPIO0)
5
OK                         # Response
AT+DIO<CR>                 # check the current DIO value
10000**000011
OK                         # Response
AT+AI<CR>                  # check the current AI value
****,****,2EE0,2EE0
OK                         # Response
```

## 8.6 Firmware Upload

### 8.6.1 Local Node

```
# enter the bootloader menu for uploading firmware. Note that UART of local
node should be set to 115200-8-N-1-N.
AT+BOOTLOAD<CR>
Press Enter…


EM250 Bootloader v20 b09
1. upload ebl
2. run
```

```
3. ebl info
BL >                          # press “1” and upload ebl file by X-modem
begin upload
CCCCCCCCCCCC
Serial upload complete        # press “2” or reset after upload is done
```

### 8.6.2 Remote Node

```
# upload firmware to a remote node using 64-bit long address. Note that UART
of local node should be set to 115200-8-N-1-N.
AT+BOOTLOAD=1,00019500002FDC40<CR>
Please start .ebl upload image ...
CCCCCCCCCCCC                  # upload ebl file by X-modem
Bootload Complete!            # succeed to upload
```

### 8.6.3 Clone

```
# clone the firmware of the local node to a remote node using 64-bit long
address. Note that UART of local node should be set to 115200-8-N-1-N.
AT+BOOTLOAD=2,00019500002FDC40<CR>
Start cloning ...
Bootload Complete!            # succeed to upload
```

# 9 S-Registers

| Register number | Descriptions |
|---|---|
| 11 | Display incoming message in command mode |
| 12 | Display command echo |
| 13 | Display command responses |
| 14 | Manage LED status |
| 21 | Inter character timeout |
| 22 | Permit joining timeout |
| 23 | TX power mode |
| 31 | Enable application-ACK |
| 32 | Transmission retries |
| 33 | Transmission timeout |
| 34 | Indirect transmission timeout |
| 35 | Broadcast range |
| 36 | Number of hops for multicast/broadcast |
| 37 | Number of hops of nonmembers for multicast |
| 38 | Receive local message for multicast/broadcast |
| 39 | Receive message by own data mode |
| 41 | GPIO pull-up |
| 42 | GPIO monitoring period |
| 43 | GPIO monitoring node |
| 44 | GPIO change detection sampling |
| 45 | GPIO remote control |
| 51 | Poll timeout for a parent node |
| 52 | Poll period for a sleepy node |
| 53 | Sleep duration |
| 54 | Standby timeout |
| 55 | Rejoin timeout |
| 56 | Poll period for an end-device |
| 61 | Sink advertise period |
| 62 | Many to one route request period |

## 9.1 Display Options

### 9.1.1 S11

| | | |
|---|---|---|
| Description | Get/Set the display mode. It determines to display incoming message in command mode | |
| Execute | ATS11?<CR> | ATS11=<value><CR> |

| | | |
|---|---|---|
| Response | <value><CR><br>OK<CR> | OK<CR> |
| Default | 0 | |
| Range | 0 - 1 | |

#### 9.1.2 S12

| | | |
|---|---|---|
| Description | Get/Set the command echo mode. | |
| Execute | ATS12?<CR> | ATS12=<value><CR> |
| Response | <value><CR><br>OK<CR> | OK<CR> |
| Default | 1 | |
| Range | 0 – 1 | |

#### 9.1.3 S13

| | | |
|---|---|---|
| Description | Get/Set the command responses mode. | |
| Execute | ATS13?<CR> | ATS13=<value><CR> |
| Response | <value><CR><br>OK<CR> | OK<CR> |
| Default | 1 | |
| Range | 0 - 1 | |

#### 9.1.4 S14

| | | |
|---|---|---|
| Description | Get/Set the LED mode. It causes the LED operation if the device is ZS10 or ZE10 set GPIO1, 2 to specific functionality.<br><br>0: LEDs are always OFF regardless of configuration or status.<br>1: LEDs are OFF when the node is under sleep status.<br>2: LEDs always operate by the case. | |
| Execute | ATS14?<CR> | ATS14=<value><CR> |
| Response | <value><CR><br>OK<CR> | OK<CR> |
| Default | 1 | |
| Range | 0 - 2 | |

## 9.2 Network Settings

#### 9.2.1 S21

| | | |
|---|---|---|
| Description | Get/Set the inter character timeout in milliseconds. It determines to send data packet after this time. | |
| Execute | ATS21?<CR> | ATS21=<value><CR> |

| | | |
|---|---|---|
| Response | <value><CR><br>OK<CR> | OK<CR> |
| Default | 100 | |
| Range | 0 - 10000 | |

#### 9.2.2 S22

| | | |
|---|---|---|
| Description | Get/Set the timeout to permit joining. It determines the time in seconds that the device will allow other devices to join to it when it starts network but an end-device cannot perform it. If set to 255, the device will always permit joining. | |
| Execute | ATS22?<CR> | ATS22=<value><CR> |
| Response | <value><CR><br>OK<CR> | OK<CR> |
| Default | 60 | |
| Range | 0 - 255 | |

#### 9.2.3 S23

| | | |
|---|---|---|
| Description | Get/Set the power boost mode of the module. If enabled, boost mode improves sensitivity by 1dB and increases output power by 2dB, improving the link margin and range. | |
| Execute | ATS23?<CR> | ATS23=<value><CR> |
| Response | <value><CR><br>OK<CR> | OK<CR> |
| Default | 0 | |
| Range | 0 – 1 | |

### 9.3 Data Transmissions

#### 9.3.1 S31

| | | |
|---|---|---|
| Description | Get/Set the application-ACK mode. If it is enabled, a node retransmits data packet when it is failed to send. A node sends next packet after receiving ACK from receiver node. This value must be set the same on the source and destination node for data transmission. | |
| Execute | ATS31?<CR> | ATS31=<value><CR> |
| Response | <value><CR><br>OK<CR> | OK<CR> |
| Default | 1 | |
| Range | 0 - 1 | |

#### 9.3.2 S32

| | |
|---|---|
| Description | Get/Set the transmission retries. If application-ACK is enabled, a sender tries to |

| | | |
|---|---|---|
| | retransmit up to this value. | |
| Execute | ATS32?<CR> | ATS32=<value><CR> |
| Response | <value><CR><br>OK<CR> | OK<CR> |
| Default | 5 | |
| Range | 0 – 99 | |

#### 9.3.3 S33

| | | |
|---|---|---|
| Description | Get/Set the transmission timeout in milliseconds. If application-ACK is enabled, a sender retransmits data packet after this time. | |
| Execute | ATS33?<CR> | ATS33=<value><CR> |
| Response | <value><CR><br>OK<CR> | OK<CR> |
| Default | 500 | |
| Range | 0 – 10000 | |

#### 9.3.4 S34

| | | |
|---|---|---|
| Description | Get/Set the indirect transmission timeout in milliseconds. It is the maximum amount of time that the MAC will hold a message for indirect transmission to a child. | |
| Execute | ATS34?<CR> | ATS34=<value><CR> |
| Response | <value><CR><br>OK<CR> | OK<CR> |
| Default | 7680 | |
| Range | 0 – 30000 | |

#### 9.3.5 S35

| | | |
|---|---|---|
| Description | Get/Set the broadcast range. ZigBee specifies three different broadcast addresses that reach different collections of nodes. Broadcasts are normally sent only to routers. Broadcasts can also be forwarded to end devices, either all of them or only those that do not sleep. Broadcasting to end devices is both significantly more resource-intensive and significantly less reliable than broadcasting to routers.<br><br>0: send to all routers<br>1: send to all non-sleepy nodes<br>2: send to all nodes in the network | |
| Execute | ATS35?<CR> | ATS35=<value><CR> |
| Response | <value><CR><br>OK<CR> | OK<CR> |
| Default | 2 | |

| | |
|---|---|
| Range | 0 – 2 |

#### 9.3.6 S36

| | | |
|---|---|---|
| Description | Get/Set the maximum number of hops the message will be relayed. The message will be delivered to all nodes within this number of hops of the sender. A value of zero is converted to maximum value(30). It is used to multicast/broadcast. | |
| Execute | ATS36?<CR> | ATS36=<value><CR> |
| Response | <value><CR><br>OK<CR> | OK<CR> |
| Default | 0 | |
| Range | 0 – 10 | |

#### 9.3.7 S37

| | | |
|---|---|---|
| Description | Get/Set the number of hops that the message will be forwarded by devices that are not members of the group. A value of zero, 7 or greater is treated as infinite. There is no longer a separate limited multicast API call. | |
| Execute | ATS37?<CR> | ATS37=<value><CR> |
| Response | <value><CR><br>OK<CR> | OK<CR> |
| Default | 0 | |
| Range | 0 – 7 | |

#### 9.3.8 S38

| | | |
|---|---|---|
| Description | Get/Set the receive message mode. If it is enabled, a local node will receive multicast or broadcast messages from itself. | |
| Execute | ATS38?<CR> | ATS38=<value><CR> |
| Response | <value><CR><br>OK<CR> | OK<CR> |
| Default | 0 | |
| Range | 0 – 1 | |

#### 9.3.9 S39

| | | |
|---|---|---|
| Description | If S39 is enabled, a local node will only receive the same type of messages with its transmit mode. For example, if local node's transmission mode is broadcast mode (AT+TM=3), it will only receive broadcast messages. | |
| Execute | ATS39?<CR> | ATS39 =<value><CR> |
| Response | <value><CR><br>OK<CR> | OK<CR> |
| Default | 0 | |

| | |
|---|---|
| Range | 0 - 1 |

## 9.4 GPIO

### 9.4.1 S41

| | | |
|---|---|---|
| Description | Get/Set the GPIO pull up/down option. If GPIOs are set to DI, it is possible to pull-up or pull-down with this value. Pull-down = 0, Pull-up = 1. | |
| Execute | ATS41?<CR> | ATS41=<value><CR> |
| Response | <value><CR><br>OK<CR> | OK<CR> |
| Default | 0 | |
| Range | 0 - 1 | |

### 9.4.2 S42

| | | |
|---|---|---|
| Description | Get/Set the GPIO monitoring period in seconds. If it is enabled, a node sends GPIO values to a remote node based on preconfigured destination IEEE address periodically and it is possible to monitor the GPIO status at the remote node. If set to 0, GPIO monitoring is disabled. | |
| Execute | ATS42?<CR> | ATS42=<value><CR> |
| Response | <value><CR><br>OK<CR> | OK<CR> |
| Default | 0 | |
| Range | 0 – 10000 | |

### 9.4.3 S43

| | | |
|---|---|---|
| Description | Get/Set the GPIO monitoring node. If S43 is 0, the monitoring node is selected by AT+DESTLA or AT+DL command. If S43 is 1, the sink node is set as the monitoring node. | |
| Execute | ATS43?<CR> | ATS43=<value><CR> |
| Response | <value><CR><br>OK<CR> | OK<CR> |
| Default | 0 | |
| Range | 0 – 10000 | |

### 9.4.4 S44

| | | |
|---|---|---|
| Description | Get/Set the GPIO change detection sampling mode. If it is enabled, a node sends GPIO values to a remote node based on preconfigured destination IEEE address immediately when the status of GPIO changes. If set to 0, GPIO monitoring is disabled. | |
| Execute | ATS44?<CR> | ATS44=<value><CR> |

| | | |
|---|---|---|
| Response | <value><CR><br>OK<CR> | OK<CR> |
| Default | 0 | |
| Range | 0 – 1 | |

#### 9.4.5 S45

| | | |
|---|---|---|
| Description | Get/Set the GPIO remote control mode. If any GPIO pin is set to digital input, a node can control the I/O status of a remote node if the I/O pin of the node is set to digital output. It is possible to use when GPIO monitoring is enabled. | |
| Execute | ATS45?<CR> | ATS45=<value><CR> |
| Response | <value><CR><br>OK<CR> | OK<CR> |
| Default | 0 | |
| Range | 0 – 1 | |

### 9.5 End Device

#### 9.5.1 S51

| | | |
|---|---|---|
| Description | Get/Set the poll timeout for a parent node in seconds. If an end device or a sleepy end device does not send a poll request to its parent coordinator or router within the poll timeout, the end device is removed from the child table. | |
| Execute | ATS51?<CR> | ATS51=<value><CR> |
| Response | <value><CR><br>OK<CR> | OK<CR> |
| Default | 60 | |
| Range | 0 – 4177920 | |

#### 9.5.2 S52

| | | |
|---|---|---|
| Description | Get/Set the sleepy end device poll period in milliseconds. Adaptive polling may allow the sleepy end device to poll more rapidly for a short time when receiving RF data. | |
| Execute | ATS52?<CR> | ATS52=<value><CR> |
| Response | <value><CR><br>OK<CR> | OK<CR> |
| Default | 100 | |
| Range | 0 - 10000 | |

#### 9.5.3 S53

| | |
|---|---|
| Description | Get/Set the sleep duration in quarter seconds. An end-device sleeps at this time and turns off radio every sleep cycle. If a node sleeps over poll timeout of parent node, the device will be removed from the child table. |

| | | |
|---|---|---|
| Execute | ATS53?<CR> | ATS53=<value><CR> |
| Response | <value><CR><br>OK<CR> | OK<CR> |
| Default | 8 | |
| Range | 0 – 16711680 | |

#### 9.5.4 S54

| | | |
|---|---|---|
| Description | Get/Set the standby timeout in seconds. When an end-device wakes up and polls with the parent node, the node will wait for another data from the ZigBee network or from the serial port for S54 before it goes to sleep mode again. If any serial or RF data is sent or received, the timer will reset and wait for next again. | |
| Execute<br><br>Response | ATS54?<CR><br><value><CR><br>OK<CR> | ATS54=<value><CR><br><br>OK<CR> |
| Default | 5 | |
| Range | 0 – 10000 | |

#### 9.5.5 S55

| | | |
|---|---|---|
| Description | Get/Set the rejoin timeout. If it is enabled, an end-device tries to rejoin the current network when contact with the network has been lost for setting value in seconds. The most common usage case is when an end device can no longer communicate with its parent and wishes to find a new one. | |
| Execute | ATS55?<CR> | ATS55=<value><CR> |
| Response | <value><CR><br>OK<CR> | OK<CR> |
| Default | 10 | |
| Range | 0 – 10000 | |

#### 9.5.6 S56

| | | |
|---|---|---|
| Description | Get/Set the end device poll period in seconds. Adaptive polling may allow the end device to poll more rapidly for a short time when receiving RF data. | |
| Execute | ATS56?<CR> | ATS56=<value><CR> |
| Response | <value><CR><br>OK<CR> | OK<CR> |
| Default | 5 | |
| Range | 0 - 10000 | |

## 9.6 Sink

### 9.6.1 S61

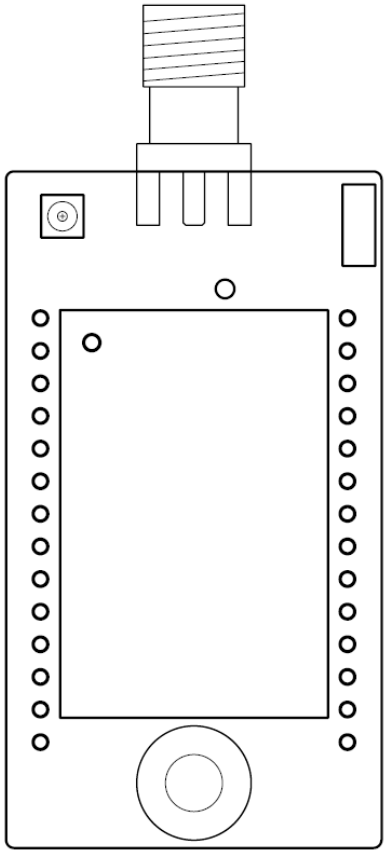
| | | |
|---|---|---|
| Description | Get/Set the sink node and the sink advertise period in seconds. When a node is defined as a sink, it can multicast its address to the rest of the network. If set to 0, sink advertise is disabled. | |
| Execute | ATS61?<CR> | ATS61=<value><CR> |
| Response | <value><CR><br>OK<CR> | OK<CR> |
| Default | 0 | |
| Range | 0 – 10000 | |

### 9.6.2 S62

| | | |
|---|---|---|
| Description | Get/Set the period to send Many-to-One Route Reqest. Many-to-One Route Request (MTOR) creates routes from every node in the network back to this node. It is useful when this node acts as a sink node. | |
| Execute | ATS62?<CR> | ATS62=<value><CR> |
| Response | <value><CR><br>OK<CR> | OK<CR> |
| Default | 0 | |
| Range | 0 – 10000 | |

# 10 Pin Assignments

| NO | Pin Assignment |
|---|---|
| 1 | GND |
| 2 | NC |
| 3 | GPIO_1 |
| 4 | GPIO_2 |
| 5 | GPIO_0 |
| 6 | Factory_Reset |
| 7 | GPIO_3 / ADC_0 |
| 8 | GPIO_4 / ADC_1 |
| 9 | H/W_Reset |
| 10 | GPIO_5 / ADC_2 |
| 11 | GPIO_6 / ADC_3 |
| 12 | +3V3 |
| 13 | NC |
| 14 | GND |



| Pin Assignment | NO |
|---|---|
| GND | 28 |
| NC | 27 |
| NC | 26 |
| NC | 25 |
| GPIO_7 | 24 |
| GPIO_9 / RTS | 23 |
| GPIO_8 / CTS | 22 |
| UART_TXD | 21 |
| UART_RXD | 20 |
| GPIO_10 / DTR | 19 |
| GPIO_11 / DSR | 18 |
| GPIO_12 | 17 |
| GND | 16 |
| GND | 15 |

*Figure 10-1 Pin Assignments*

*Table 10-1 Pin Assignments Table*

| Pin | NAME | Default Function | Direction | Description |
|---|---|---|---|---|
| 1 | GND | - | - | Ground |
| 2 | NC | - | - | Not Connect |
| 3 | GPIO_1 | DIO_1 | IN/OUT | Digital I/O, Power LED |
| 4 | GPIO_2 | DIO_2 | IN/OUT | Digital I/O, Status LED |
| 5 | GPIO_0 | DIO_0 | IN/OUT | Digital I/O, Permit joining input |
| 6 | Factory Reset | Factory Reset | IN | Digital I/O, Factory reset input |
| 7 | GPIO_3 | DIO_3 | IN/OUT | Digital I/O / ADC_0 |
| 8 | GPIO_4 | DIO_4 | IN/OUT | Digital I/O / ADC_1 |
| 9 | H/W_/Reset | - | IN | Reset, Active low, > 5ms to cause a reset |
| 10 | GPIO_5 | DIO_5 | IN/OUT | Digital I/O / ADC_2 |
| 11 | GPIO_6 | DIO_6 | IN/OUT | Digital I/O / ADC_3 |
| 12 | +3V3 | - | IN | Power supply for system, 3.3V |
| 13 | NC | - | - | Not Connect |
| 14 | GND | - | - | Ground |
| 15 | GND | - | - | Ground |
| 16 | GND | - | - | Ground |
| 17 | GPIO_12 | DIO_12 | IN/OUT | Digital I/O |
| 18 | GPIO_11 | DIO_11 | IN/OUT | Digital I/O, UART_DSR |
| 19 | GPIO_10 | DIO_10 | IN/OUT | Digital I/O, UART_DTR |
| 20 | UART_RXD | UART_RXD | IN | UART Data Input |
| 21 | UART_TXD | UART_TXD | OUT | UART Data Output |
| 22 | GPIO_8 | DIO_8 | IN/OUT | Digital I/O, UART_CTS |
| 23 | GPIO_9 | DIO_9 | IN/OUT | Digital I/O, UART_RTS |
| 24 | GPIO_7 | DIO_7 | IN/OUT | Digital I/O |
| 25 | NC | - | - | Not Connect |
| 26 | NC | - | - | Not Connect |
| 27 | NC | - | - | Not Connect |
| 28 | GND | - | - | Ground |

# 11 Mechanical Drawings

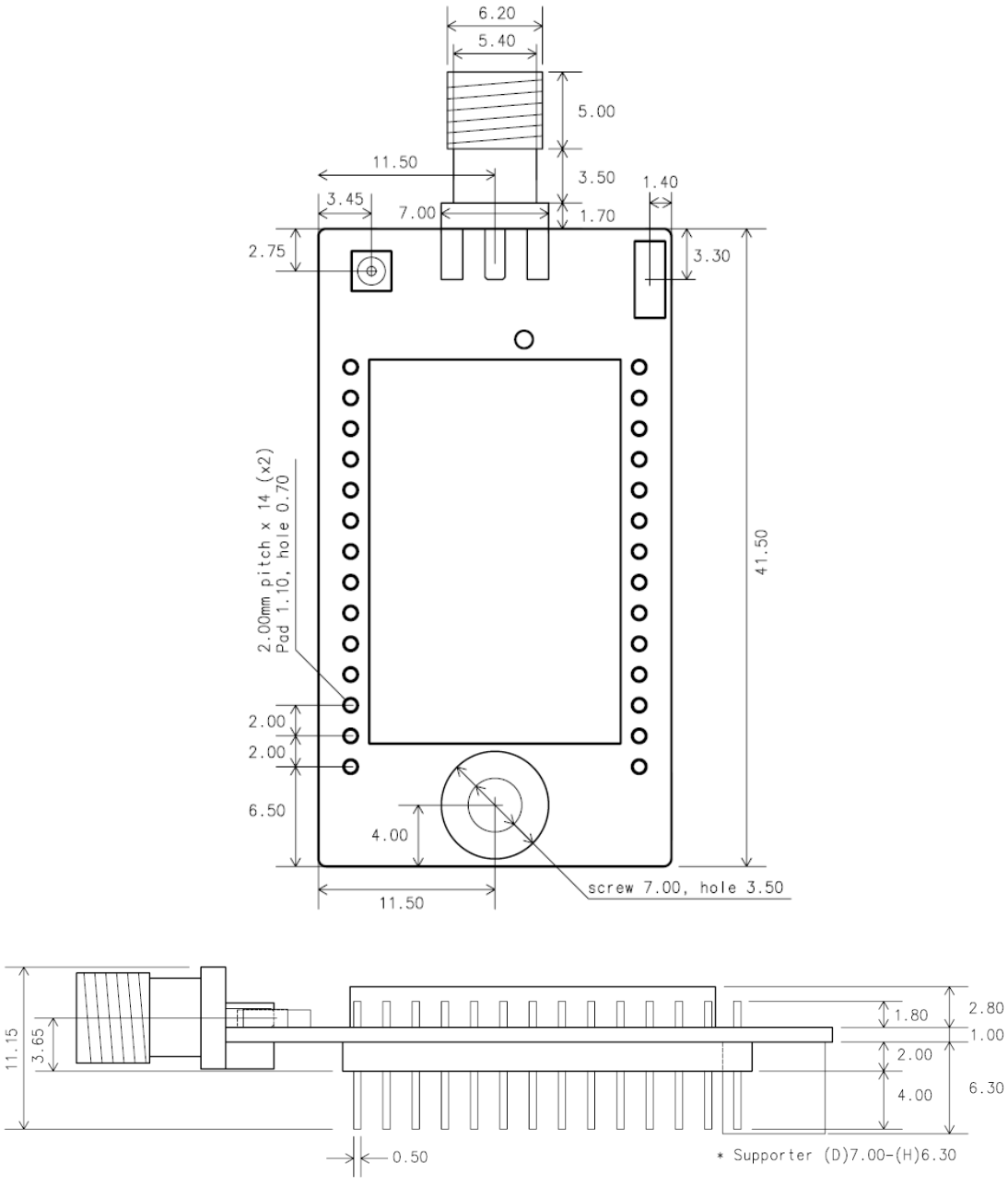



*Figure 11-1 Mechanical Drawings*

# 12 Warranty

## 12.1 GENERAL WARRANTY POLICY

Sena Technologies, Inc. (hereinafter referred to as SENA) warrants that the Product shall conform to and perform in accordance with published technical specifications and the accompanying written materials, and shall be free of defects in materials and workmanship, for the period of time herein indicated, such warranty period commencing upon receipt of the Product.

This warranty is limited to the repair and/or replacement, at SENA's discretion, of defective or non-conforming Product, and SENA shall not be responsible for the failure of the Product to perform specified functions, or any other non- conformance caused by or attributable to: (a) any misapplication or misuse of the Product; (b) failure of Customer to adhere to any of SENA's specifications or instructions; (c) neglect of, abuse of, or accident to, the Product; or (d) any associated or complementary equipment or software not furnished by SENA.

Limited warranty service may be obtained by delivering the Product to SENA or to the international distributor it was purchased through and providing proof of purchase or receipt date. Customer agrees to insure the Product or assume the risk of loss or damage in transit, to prepay shipping charges to SENA, and to use the original shipping container or equivalent.

## 12.2 LIMITATION OF LIABILITY

EXCEPT AS EXPRESSLY PROVIDED HEREIN, SENA MAKES NO WARRANTY OF ANY KIND, EXPRESSED OR IMPLIED, WITH RESPECT TO ANY EQUIPMENT, PARTS OR SERVICES PROVIDED PURSUANT TO THIS AGREEMENT, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. NEITHER SENA NOR ITS DEALER SHALL BE LIABLE FOR ANY OTHER DAMAGES, INCLUDING BUT NOT LIMITED TO DIRECT, INDIRECT, INCIDENTAL, SPECIAL OR CONSEQUENTIAL DAMAGES, WHETHER IN AN ACTION IN CONTRACT OR TORT (INCLUDING NEGLIGENCE AND STRICT LIABILITY), SUCH AS, BUT NOT LIMITED TO, LOSS OF ANTICIPATED PROFITS OR BENEFITS RESULTING FROM, OR ARISING OUT OF, OR IN CONNECTION WITH THE USE OF FURNISHING OF EQUIPMENT, PARTS OR SERVICES HEREUNDER OR THE PERFORMANCE, USE OR INABILITY TO USE THE SAME, EVEN IF SENA OR ITS DEALER HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. IN NO EVENT WILL SENA OR ITS DEALERS TOTAL LIABILITY EXCEED THE PRICE PAID FOR THE PRODUCT.

## 12.3 HARDWARE PRODUCT WARRANTY DETAILS

WARRANTY PERIOD: SENA warranties embedded hardware Product for a period of one (1) year, and

external hardware Product for a period of three (3) or five (5) years according to the Product type.

WARRANTY PROCEDURE: Upon return of the hardware Product SENA will, at its option, repair or replace Product at no additional charge, freight prepaid, except as set forth below. Repair parts and replacement Product will be furnished on an exchange basis and will be either reconditioned or new. All replaced Product and parts become the property of SENA. If SENA determines that the Product is not under warranty, it will, at the Customers option, repair the Product using current SENA standard rates for parts and labor, and return the Product at no charge in or out of warranty.

WARRANTY EXCLUSIONS: Damages caused by
- Accidents, falls, objects striking the SENA product,
- Operating the Product in environments that exceed SENA's temperature and humidity specifications,
- Power fluctuations, high voltage discharges,
- Improper grounding, incorrect cabling,
- Misuse, negligence by the customer or any other third party,
- Failure to install or operate the product (s) in accordance to their SENA User Manual,
- Failure caused by improper or inadequate maintenance by the customer or any other third party,
- Floods, lightning, earthquakes,
- Water spills,
- Replacement of parts due to normal wear and tear,
- Hardware has been altered in any way,
- Product that has been exposed to repair attempts by a third party without SENA's written consent,
- Hardware hosting modified SENA Software, or non-SENA Software, unless modifications have been approved by Sena.
- Battery component capacity degradation due to usage, aging, and with some chemistry, lack of maintenance.

## 12.4 SOFTWARE PRODUCT WARRANTY DETAILS

WARRANTY PERIOD: SENA warranties software Product for a period of one (1) year.

WARRANTY COVERAGE: SENA warranty will be limited to providing a software bug fix or a software patch, at a reasonable time after the user notifies SENA of software non-conformance.

## 12.5 THIRD-PARTY SOFTWARE PRODUCT WARRANTY DETAILS

The warranty policy of the third-party software is conformed to the policy of the corresponding vendor.

# 13 Regulatory Information

## 13.1 FCC

FCC Rule: Part 15 Subpart C Section 15.247
FCCID: S7AZBLINXZE10

**FCC Compliance Statement**

This device complies with part 15 of the FCC Rules. Operation is subject to the following two conditions:
(1) This device may not cause harmful interference, and
(2) This device must accept any interference received,
Including interference that may cause undesired operation

**Information to User**

This equipment has been tested and found to comply with limits for a Class B digital device, Pursuant to Part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference in a residential installation.
This equipment generate, uses and can radiate radio frequency energy and, if not installed and used in accordance with the instructions, may cause harmful interference to radio communications.
However, there is no guarantee that interference will not occur in a particular installation. If this equipment does cause harmful interference to radio or television reception, which can be determined by turning the equipment off and on, the user is encouraged to try to correct the interference by on or more of the following measures:

- Reorient or relocate the receiving antenna.
- Increase the separation between the equipment and receiver-Connect the equipment into an outlet a circuit different form that to which the receiver is connected.
- Consult the dealer or an experienced radio/TV technician for help.

**RF Exposure Statement**

The equipment complies with FCC RF radiation exposure limits set forth for an uncontrolled environment. This device and its antenna must not be co-located or operation in conjunction with any other antenna or transmitter.

**Do not**

Any changes or modifications to the equipment not expressly approved by the party responsible for compliance could void user's authority to operate the equipment.

## 13.2 CE

CE1177(!)
We, Sena Technologies, Inc., declare that the product(s):

- ProBee-ZE10 ZigBee OEM Module

to which this declaration relates is in conformity with the following standard(s) or other normative document(s)

- EN 60950-1
- ETSI EN 301 489-1
- ETSI EN 301 489-17
- ETSI EN 300 328

Following the provisions of

- EMC DIRECTIVE 1999/5/EC

## 13.3 TELEC

Technical Regulations for Specified Radio Equipment Article 2, Section 1 (19)
Certification No: 010WWBT32

## 13.4 KCC

Type Registration

Certification No: SNA-ZBlinx-ZE10

Basic Model Number: ZBlinx-ZE10S

Series Model Number: ProBee-ZE10U, ProBee-ZE10W, ProBee-ZE10C

## 13.5 ZigBee Alliance

Manufacturer Specific

ZIG10006PRV38490-24

# 14 RF Information

**Radio Frequency Range**
2.410~2.475GHz

**Number of Frequency Channel**
14 channels

**Transmission Method**
DSSS (Direct Sequence Spread Spectrum)

**Modulation Method**
O-QPSK (Offset Quadrate Phase Shift Keying)

**Radio Output Power**
+18dBm

**Receiving Sensitivity**
-102dBm

**Power Supply**
DC3.3V